VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

PROSTŘEDÍ PRO ANALÝZU ŠKODLIVÉHO KÓDU CUCKOO SANDBOX
AUTOMATED MALWARE ANALYSIS BASED ON CUCKOO SANDBOX

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Kamil Hons**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Zdeněk Martinásek, Ph.D.**
SUPERVISOR

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Informační bezpečnost**
Ústav telekomunikací

Student: Kamil Hons ***ID:*** 195834
Ročník: 3 ***Akademický rok:*** 2018/19

NÁZEV TÉMATU:

Prostředí pro analýzu škodlivého kódu Cuckoo Sandbox

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je návrh hardwarového a softwarového prostředí pro analýzu škodlivého kódu. Prostředí pro rozbor škodlivého kódu musí být navrženo způsobem, který umožní spuštění škodlivého kódu v omezeném a kontrolovaném prostředí i zdokumentování chování tohoto škodlivého kódu. Vhodným výchozím prostředím pro rozbor škodlivého kódu je např. sandbox Cuckoo, který umožňuje dynamickou analýzu chování škodlivého kódu. Zvolený sandbox a dynamickou analýzu škodlivého chování je nutné doplnit o statickou analýzu škodlivého kódu a vhodné prostředí o prostředky pro tuto analýzu např. (Remnux/CAINE). Zanalyzujte možnosti a popište požadavky na vytvoření tohoto prostředí. Prostředí navrhněte a vytvořte. Do prostředí zaneste několik škodlivých kódů a prostředí použijte pro simulovanou analýzu těchto kódu. Průběh a dosažené závěry simulace popište a prezentujte.

DOPORUČENÁ LITERATURA:

[1] PEARCE, Lauren. Malware analysis in a nutshell. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2016.

[2] GUARNIERI, Claudio, et al. The Cuckoo Sandbox (2012). URL https://www. cuckoosandbox. org, 2012.

Termín zadání: 1.2.2019 ***Termín odevzdání:*** 27.5.2019

Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.
Konzultant: Ing. Jaroslav Rus (ANECT a.s.)

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Bakalářská práce se zabývá vytvořením kontrolovaného prostředí pro analýzu škodlivého kódu, ve kterém je provedena a popsána simulovaná analýza vybraného vzorku. Dále je obsahem práce analýza webového prostředí se zaměřením na malvertising kampaně a sdílení nalezených školdivých kódů.

KLÍČOVÁ SLOVA

škodlivý kód, sandbox, malware, malvertising, analýza

ABSTRAKT

The bachelor thesis deals with the creation of a controlled environment for an analysis of malicious code, where simulated analysis of a selected sample is conducted and described. Furthermore, the content of the work is a web environment analysis aimed on malvertising campaigns and sharing founded malicious codes.

KLÍČOVÁ SLOVA

malicious code, sandbox, malware, malvertising, analysis

HONS, Kamil. *Prostředí pro analýzu škodlivého kódu Cuckoo Sandbox*. Brno, 2018, 62 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Zdeněk Martinásek, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Prostředí pro analýzu škodlivého kódu Cuckoo Sandbox“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Ing. Zdeňku Martináskovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych chtěl poděkovat konzultantantovi mé bakalářské práce panu Jaroslavu Rusovi za cenné rady, odborné vedení, praktické návrhy řešení a trpělivost při konzultacích této práce.

Brno ....

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Počítače a síťová komunikace je dnes naprosto běžná pro drtivou většinu lidí. V oblasti internetu je obrovské množství dat usnadňující lidem život. Díky dostupnosti informací v řádech sekund a komunitě, která takovéto informace vyhledává, se však v tomto prostředí otevírá prostor pro kyberkriminalitu a nepřeberné množství škodlivých kódů jako Adware (Advertising-supported software), Trojan (Trojan Horse), Ransomware, Rootkit, Spyware, Virus, Worm. Tyto kódy jsou vylepšovány a zdokonalovány tak, aby jejich detekce byla co nejsložitější. Technik k ukrývání škodlivých kódu uvnitř jiného nosiče (například aplikace), je mnoho. Základní rozdělení by však mohlo být na obfuskaci kódu a na zabalení kódu.

Těchto a mnoho dalších technik je využíváno právě v oblasti kyberkriminality za účelem lepší a efektivnější distribuce škodlivých kódů. Zejména oblast šíření malwaru prostřednictvím tzv. „drive-by-malware“ techniky, zaznamenalo velký nárůst ve spojení s malvertisingovými kampaněmi. Kombinace tak obrovské distribuční sítě, jako jsou cílené reklamy po celém internetu a technice nevyžadující žádný zásah uživatele proto, aby byl nakažen malwarem, se stává přední hrozbou v oblasti kyberkriminality.

Hlavním cílem bakalářské práce je vytvoření sandboxového prostředí pro dynamickou analýzu škodlivého kódu, ve kterém bude provedena a zdokumentována simulovaná analýza škodlivého kódu. Prostředí musí být vytvořeno s důrazem na izolovanost sebe sama od hostovského systému. A simulovanou analýzu je třeba doplnit o statickou analýzu.

Dílčím cílem bakalářské práce je také praktická analýza webového prostředí se zaměřením na malvertising s využitím předem testovaného sandboxového prostředí a jeho reálné použití v nesimulovaném prostředí, spolu s publikací možných výsledků v rámci příslušných komunit.

1 Škodlivý kód

V obecném měřítku bychom mluvili o kyber zločincích v souvislosti, která není spjatá s hmotným ziskem. Hackeři (lidé kteří protiprávně obcházejí zabezpečení zařízení) jsou tedy spíše, ne však výlučně, spojováni s motivací v podobě vzrušení, zábavy, msty nebo například zvědavosti[1, 2]. Jako typicky hackarské aktivity bychom mohli označit prolomení zabezpečení stránek a zanechání ukryté zprávy o návštěvě zabezpečených míst, bez vzniku hmotné škody provozovatele stránek. Celá otázka hranic mezi hackingem a crackingem je však mnohem složitější a z pohledu práva velmi diskutovaná.

Cracking je naopak spojován s jednoznačnou vidinou hmotného zisku, nebo alespoň nevratné škody na cílovém uživateli [1, 2, 3]. Pokud bychom měli uvést příklad crackingu, šlo by například o distributed denial of service (DDOS) útok, kdy cracker zahltí server množstvím requestů a využije nezabezpečení proti takovému útoku. Mnohdy k tomu využije síť Botnet. Tento útok povede pravděpodobně k ušlému zisku provozovatele serveru.

1.1 Druhy a účinky

Adware je druh škodlivého kódu, který uživateli předhazuje nevyžádané reklamy, běžné na webových stránkách. Může být také součástí aplikací, které následně propagují reklamy v rámci jejich používání. Dále je jeho účelem sledovat zájmy uživatele a předávat je třetím stranám pro cílenou reklamu [4]. **Trojan** je program ukrytý v rámci jiného souboru, jako příloha emailu, nebo například aplikace. Jeho účelem je získat, pokud možno nejvyšší přístup do napadeného zařízení a vykonat požadované úlohy útočníka. Mezi tyto úkony patří např. odesílání dat prostřednictvím internetu, nebo vytváření zadních vrátek umožňujících obejít systémové zabezpečení přístupu [5]. **Ransomware** je navržen tak, aby zašifroval disk napadeného počítače (nebo alespoň jeho části) a znemožnil tak čtení dat uživateli. Běžně útočník přiloží zprávu o tomto šifrování s pokyny k zaplacení částky, po jejímž uhrazení dostane oběť klíč k rozšifrování disku a získání dat zpět viz obr. 1.1. V některých případech však jde pouze o hoax [6]. Což je případ, kdy k žádnému útoku nedošlo, ale útočník se pouze snaží zmást oběť k zaslání peněz falešnou zprávou. **Rootkit** je škodlivý program, jehož účelem je získat přístup do systému a následně do něj umožnit útočníkovi neomezený přístup. Díky tomu je velmi dobře skrytý před ochrannými softwarovými prostředky [7]. **Spyware** má za úkol sledovat napadeného uživatele při jeho běžných aktivitách, může zaznamenávat vstupy klávesnice, síťovou aktivitu, nebo získat přístup k periferiím zařízení jako webkamera dále je schopen skenovat a monitorovat aktivitu napadeného systému [4]. **Virus** je sebereplikující kód, který se dokáže šířit

Obr. 1.1: Ransomware Bad Rabbit [8].

pomocí skriptů, dokumentů ale také chyb zabezpečení webových aplikací. Na infikovaných zařízeních může získávat informace, poškozovat funkcionalitu systému, podvrhovat chování aplikací, nebo například vytvářet Robot network (Botnet) [9]. **Worm** je druh škodlivého kódu který se na rozdíl od Virů replikuje bez nutnosti uživatelské interakce. Jeho účel je však stejný, získat data, narušit chování systému, začlenit hostitelské zařízení do Botnetu. Typicky využívá slabin zabezpečení aplikací a systémů [9].

1.2 Ochrané prvky

Prvky, kterými se lze před škodlivými kódy bránit, jsou zaměřeny na sledování různých částí zařízení a ochranou před napadením. Prvky které se snaží předcházet napadení jsou například: antivirus software (AVS), antispyware, firewally, štíty, aktualizační software. Další ochranu poskytují skeny odhalující proniknutí škodlivých kódů do systému.

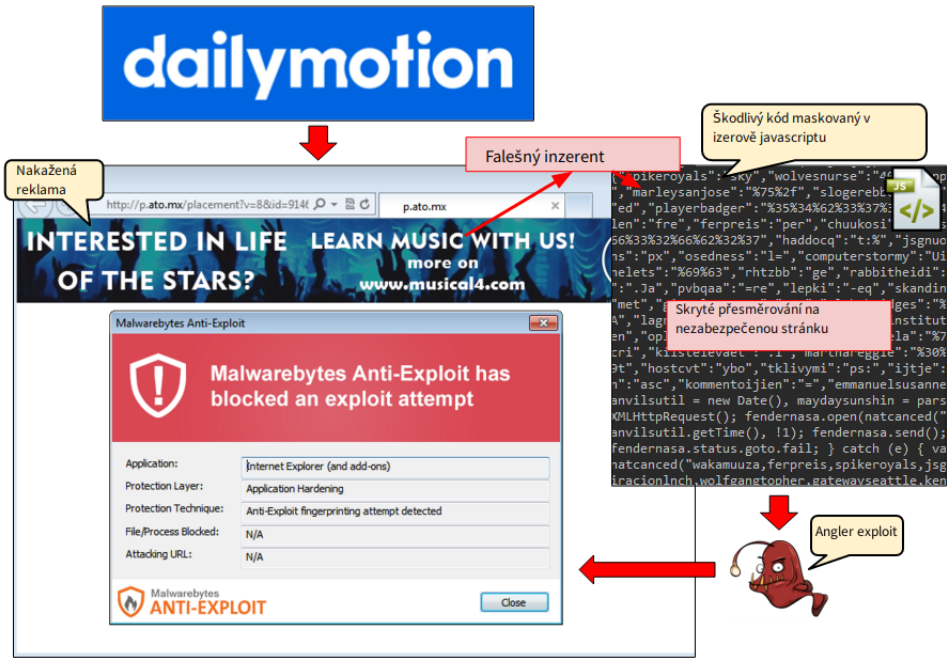
Antispyware je software pracující se stejnými technikami jako AVS, jeho zaměření detekce je ovšem pouze na spyware, jak název napovídá. Firewally mohou být fyzické nebo softwarové. Výhoda fyzických firewallů je větší možnost umístění v síti. Tyto ochranné prvky blokují síťový provoz na základě předefinovaných pravidel a předchází tak mnohým útokům právě zablokováním samotného pokusu o spojení. Jelikož mnoho škodlivých softwarů využívá chyb v zabezpečení aplikací, aktualizační software se stará o to, aby software uživatele byl co nejvíce aktuální, a tím zabránil útočníkům využití takto odhalených slabin [10]. AVS je program využívající detekčních technik k odhalení škodlivého softwaru. Jeho úspěšnost se zakládá

právě na zdokonalování a konstantním aktualizování databází, získaných na základě těchto technik. Detekce pomocí signatur je technika vytváření otisků nebo hashů škodlivých kódů, které konstantně vnáší do databáze pro následné porovnání AVS na uživatelském počítači. V případě shody otisku je program označen jako nebezpečný. Škodlivé kódy se proti této technice brání mutací kódu tak, aby nenastávaly shody. Z tohoto důvodu nastupuje heuristická analýza, která již nepotřebuje naprostou shodu otisku. Zkoumá, zda se v kódu nenachází nepatřičné instrukce. Může využívat statické analýzy, která bude v této práci popsána podrobněji. Další technikou je analýza chování kódu. Jejím úkolem je spustit podezřelý soubor a sledovat jeho chování, například zda v rámci spuštění nerozbaluje a nespouští další skryté programy. Rovněž zkoumá zda v programu nedochází k narušení integrity systému v podobě upravování souborů, nebo sledování uživatelské aktivity [11]. K tomu využívá dynamické analýzy v sandboxovém prostředí, kterou se v rámci práce budu rovněž zabývat ve větší míře. Tato analýza přibližuje AVS blíže k host intrusion prevention system (HIPS), který je využíván ke sledování aktivity programů spuštěných uživatelem.

1.3 Malvertising

Jde o způsob šíření malwaru prostřednictvím Real Time Bidding (RTB), což je aukční model zajišťující dnes všudy přítomnou cílenou reklamu na uživatele [12]. RTB slouží k aukcím v reálném čase, kde jedna strana nabízí prostor pro reklamu a strana druhá tento prostor nakupuje. Tímto procesem je zajištěno, že cílena reklamní kampaň se na základě cookies, geografické polohy, nebo jiných sdílených informací dostane ke konkrétní skupině uživatelů [12]. Tohoto procesu využívají malvaretising kampaně, které místo reklam raději šíří malware [13]. Názorně je tento proces ukázán na obrázku 1.2

Rozdělujeme tři skupiny koncového dodání malwaru. První skupinou je využití tzv. „deceptive download“, tento přístup vyzývá uživatele ke stažení rozšíření do prohlížeče, nebo softwaru k následovnému využití k napadení systému, jedná se tak o jistou formu sociálního hackingu[13]. Druhá skupina tzv. „link hijacking“ kdy je uživatel přesměrován z původní webové stránky na jinou zajišťující přenos malwaru [13]. Poslední a nejnebezpečnější skupinou je tzv. „drive-by-downlads“, v tomto případě není zapotřebí uživatelských zásahů a přenos malwaru je proveden skrze aplikace třetích stran a jejich známé slabiny[13]. Zejména jsou využity podpůrné programy pro webovou funkcionalitu. Nejznámější z nich jsou doplňky Java a Flash, je však možné využít i neaktuální prohlížeče atd.



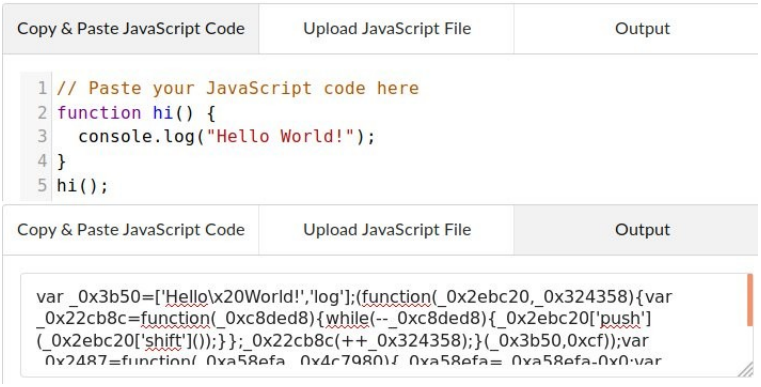
Obr. 1.2: Proces odkázání na nezabezpečený web pro stažení malwaru. [14]

1.4 Obfuskační techniky malwaru

Z důvodu rozšířenosti softwaru detekujícího malware na základě statického porovnání signatur, hashů, nebo například volaných funkcí. Je možné se setkat s protiopatřením v podobě obfuskačních technik. Jednou z možných variant maskování binárního souboru je metoda „Opaque Constants“, jedná se o zneprůhlednění použitých hodnot a konstant takovým způsobem, aby byla ztížena statická detekce. Toho je v tomto přístupu dosaženo zavedením náhodného prvku generující vlastní pole hodnot, které se bude při různých spuštěních lišit. Je tak běžné nepředávat konstantní hodnoty přímo, ale jako celou funkci generující vždy stejný výsledek [15].

Další metodou je „ControlFlow Obfuscation“, jejímž záměrem je znesnadnit analýzu jednotlivých instrukcí programu. Toho dosahuje pomocí vytváření vícero nepodmíněných volání a provádění nadbytečných funkcí v rámci svého spuštění. Při tomto procesu jsou tedy podstatné části kódu zachovány a v paměti je možné nalézt patřičné operace a volání. Nicméně sled a celkový počet operací se v různých spuštěních liší a to zejména kvůli náhodným operacím spojeným s první technikou „Opaque Constants“ [15].

Těchto technik se využívá například při ochraně JavaScript kódů a existuje mnoho online nástrojů, které pro vás váš kód upraví. Jak vypadá jednoduchá funkce „Hello World“ napsaná v JavaScriptu a následně obfuskovaná za pomoci webové stránky `https://obfuscator.io/` je na obrázku 1.3[16].



Obr. 1.3: Online nástroj pro obfuskaci kódu [16].

1.5 Dropper

Jedná se o binární soubory jejichž účelem je stažení dalšího exe souboru, ten může být finálním payloadem, nebo také dalším dropperem [17]. Důvodem použití dropperu ještě před stažením finálního payloadu, je zmapování cílového systému. Toho je možné dosáhnout čtením některých Windows Registrů nebo například nalezením řetězce poukazující na využití prostředků virtuálního prostředí [18]. Cílem těchto předeslaných dropperů je odhalit, zda místo ve kterém se nachází, není virtuální operační systém. Tyto informace jsou schopny vyhodnotit a následně rozhodnout o stažení dalších souborů. Některé speciální droppery byly vytvořeny pro distribuci různých malwarů od rozdílných rodin, díky infrastruktuře Pay Per Install (PPI) je možné cílit nasazení malwaru s určitým zaměřením, například na základě geografické polohy.

2 Sandboxové prostředí

Na internetu je dostupné množství softwaru zabývající se problematikou analyzování škodlivého kódu. Mezi hlavní zástupce patří JoeSandbox, Falcon Sandbox, FireEye Sandbox, nebo Cuckoo Sandbox. Důvodem, proč používáme sandbox je sjednocení nástrojů pro analýzu škodlivých souborů stejně tak jako vytvoření bezpečného prostředí pro tyto úkony. Základním prvkem sandboxu je vytvoření virtuálního prostředí. Pro tento účel můžeme využít virtualizačního softwaru jako Kernel-based Virtual Machine (KVM), WMware, nebo Oracle VM VirtualBox.

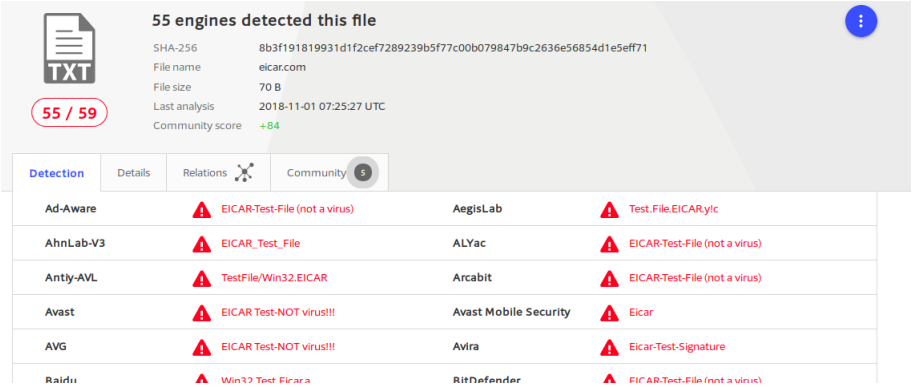
Virtuální prostředí je nezbytné z hlediska bezpečnosti, počítač sdílí hardwarové prostředky jako procesor nebo operační paměť. Předejití nedovolenému čtení (memory dump), lze přidělením hardwarových prostředků virtuálnímu operačnímu systému, díky čemuž nelze číst paměť hostujícího počítače. Sandbox, jakožto nástroj pro automatickou malwarovou analýzu, usnadňuje využití mnoha nástrojů pro reverzní inženýrství. Statickou analýzu s výpisem volání funkcí a knihoven škodlivého souboru. Dynamickou analýzu shrnující chování programu v přehledném výstupu. Sandbox odhaluje kam se program chtěl připojit, jaké informace sdílel a jaké přijal, kam uložil soubory, jaké soubory modifikoval. Automaticky porovnává signatury souboru pro případ, že se jedná o známý malware [19, 20].

2.1 Statická analýza

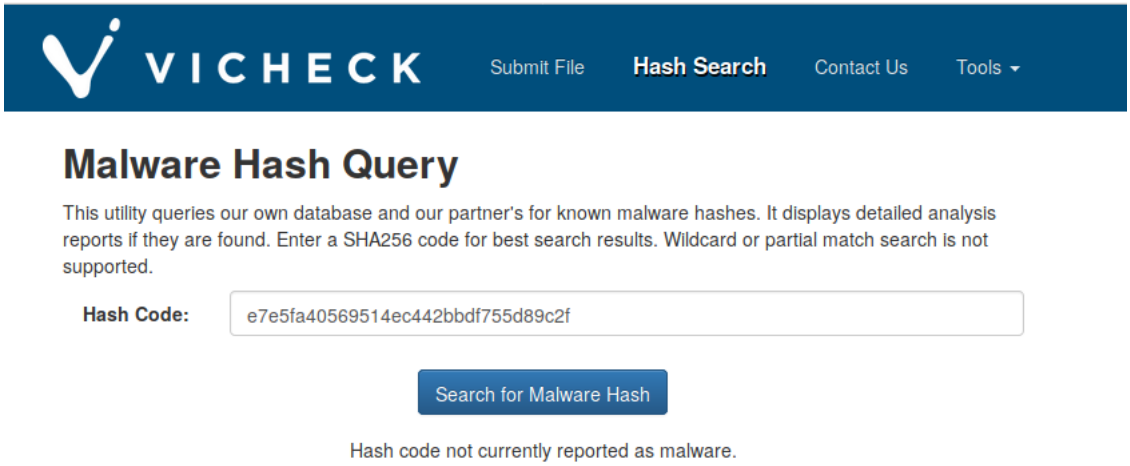
Základním aspektem statické analýzy je studování struktury kódu. Nedochází při ní ke spuštění samotného programu [21]. První variantou této analýzy je využití skenu různých AVS. Základem je široká databáze známých virů, ve které je možné hledat a porovnávat zvolený soubor. Nevýhoda tohoto postupu spočívá v nových virech, které v těchto databázích obsaženy nejsou. Využití antivirových skenů je možné například přes stránky jako VirusTotal viz Obr. 2.1, která sdružuje hned několik antivirových databází [21].

Další variantou statické analýzy, je porovnání na základě hashe. Ten je spočítán některým z hashovacích algoritmů jako například MD5, ze souboru. Následně můžeme tento hash porovnávat s dostupnými hashy online například přes vicheck portál jako na Obr. 2.2 [21]. Tento postup je v této práci také prakticky využit a demonstrován pro porovnání otisku systémové části virtuálního disku operačního systému Windows 7.

Poslední variantou statické analýzy je prohledávání souboru pro výskyt řetězců. Pokud je program škodlivý, lze předpokládat, že bude chtít komunikovat s nějakým serverem nebo bude upravovat konkrétní složky. Díky výpisu řetězců programu můžeme takové úmysly odhalit [21]. V těchto případech se však mnohdy setkáváme se



Obr. 2.1: Výsledek skenu eicar souboru [22].



Obr. 2.2: Porovnávání hashe s databází vicheck [23].

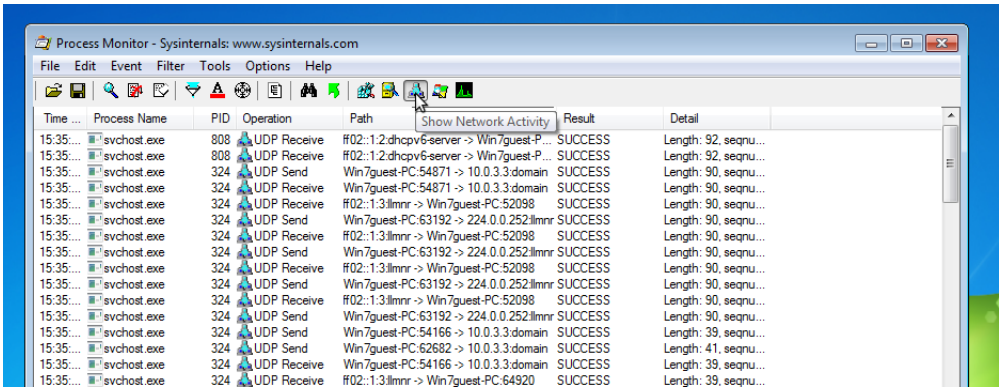
skrýváním škodlivé části kódu. Důležité soubory viru, včetně řetězců, jsou zabaleny v nečitelné podobě (komprimovány, nebo šifrovány). Takto zabalený program s šifrovaným, neznámým obsahem, obsahuje méně řetězců a musí zahrnovat knihovny pro samotné rozbalení. To může být zásadním znakem pro škodlivý kód [21]. Protože statická analýza žádný kód nespouští, dostáváme se na konec jejích možností.

2.2 Dynamická analýza

Této analýzy se využívá ke spuštění škodlivého kódu, které může vést k projevení vlastností, jež statická analýza odhalit nemůže. Nejvhodnějším řešením pro bezpečné spuštění malwaru, je využití virtuálního prostředí. To zajistí, aby se nic nedostalo z kontrolovaného systému na počítač hostitele. Problémem však může být chybějící

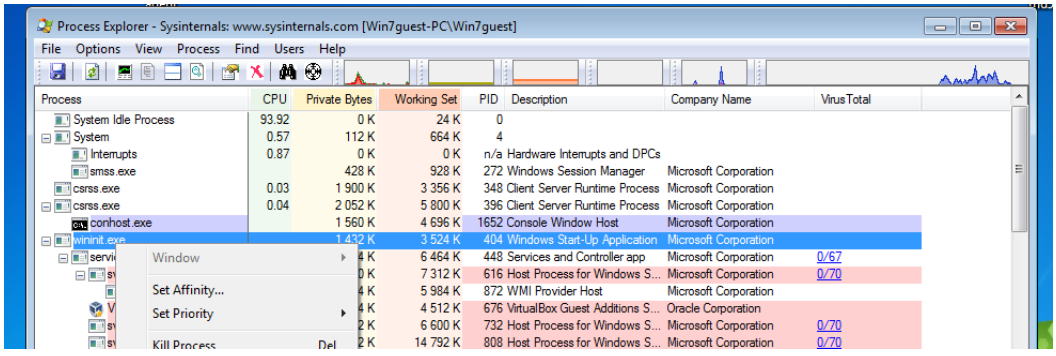
internetové připojení, z důvodu izolace virtuálního prostředí, nebo samotný fakt, že některý malware dokáže detekovat spuštění sebe sama ve virtuálním prostředí.

Dynamická analýza obnáší monitorování spuštěných procesů. Nemusí vždy zachytit všechny procesy, jako například některé systémové volaní Graphical User Interface (GUI). Programy jako Process Monitor, v případě platformy windows, jsou schopné monitorovat síťovou aktivitu procesů (viz obr. 2.3) s konkrétním portem pro komunikaci, jakékoli další spuštěné procesy, zásahy do souborového systému nebo způsob, jakým byl kód nainstalován v registru [21].



Obr. 2.3: Zobrazení síťové aktivity pro proces svchost.exe.

Zkoumání procesů programy jako Process Explorer společnosti Microsoft umožňuje pro každý běžící proces výpis načtených knihoven, zjištění rodičovského procesu a podrobnosti jako jméno, process identifier (PID), popis, nebo jméno vydavatele. Součástí je také možnost ověření pravosti binárního souboru na disku proti databázi windows otisků. Je možné ověřit, zda je soubor opravdu spuštěn systémem či nikoliv. Dále můžeme ověřit spuštěné procesy integrovanou funkcí programu na stránkách VirusTotal [22] viz obrázek 2.4.



Obr. 2.4: Testování procesů stránkou virus total v programu Process Explorer [22].

Škodlivý kód mnohdy využívá záměnu procesů. Při tomto postupu je originální proces zkopírován avšak pouze proto, aby prošel kontrolou otisku, navzdory tomu, že v operační paměti již běží naprosto jiný program. To se projevuje například deseti spuštěnými procesy explorer.exe [21]. Při dynamické analýze se však můžeme podívat na řetězce a zjistit zásadní odlišnosti v tom, co používá zkopírovaný proces oproti původnímu procesu na disku [21].

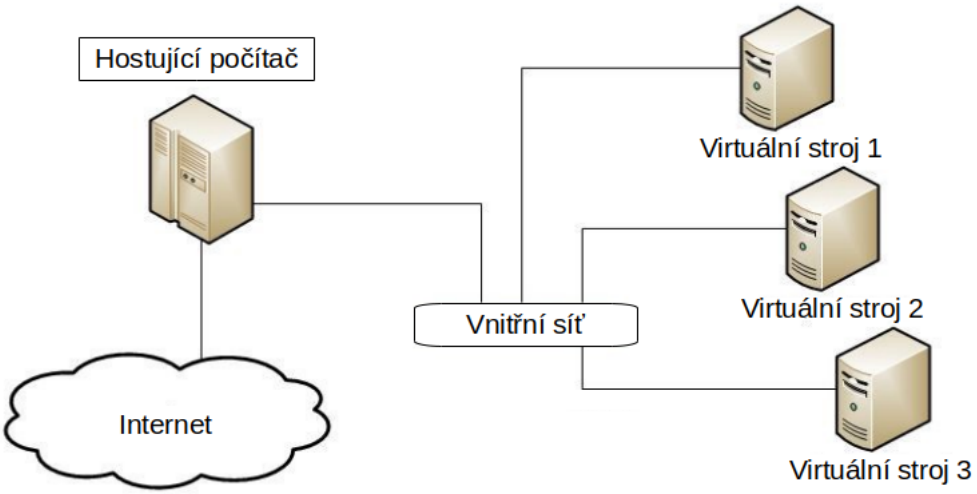
K porovnávání otisků registrů slouží programy md5deep, Regshot, RegistryChnagesView a mnoho podobných, umožňuje rekurzivně vytvořit otisky souborů na disku, uchovávat je a porovnávat. Jedna sada otisků vytvořena před spuštěním škodlivého souboru, je porovnána se sadou vytvořenou po spuštění. Tímto porovnáním poté zjistíme modifikace registrů nebo jiných částí systému.

Podvržení síťového spojení je mnohdy nezbytné proto, aby binární soubor skutečně provedl útok. Některé škodlivé kódy bez síťových instrukcí vzdáleného serveru nefungují a nelze tak dynamickou analýzou odhalovat jejich účel. Aplikace jako ApateDNS, nebo INetSim jsou navrženy tak, aby pro spuštěný binární soubor vyžadující internetovou komunikaci vyhověli, bez bezpečnostních rizik. ApateDNS umožňuje zjištění doménových jmen vzdálených serverů, se kterými chce škodlivý kód komunikovat. INetSim simuluje internetové služby jako Trivial File Transfer Protocol (TFTP), Post Office Protocol (POP3), Hypertext Transfer Protocol (HTTP). Aplikace podporuje také funkci faketime, díky níž může INetSim podvrhnout komunikaci v jakémkoli čase. Účelem je například vynucení denního či nočního módu komunikace škodlivého souboru. Součástí je logování veškerých paketů v obou směrech a následné shrnutí dat [24] [21].

3 Cuckoo Sandbox

Jedná se o opensource software pro automatickou malwarovou analýzu. Může být využit pro analýzu spustitelných souborů, pdf, emailů, webových stránek, archivů, atd. Dokáže zaznamenávat použité knihovny Application Programming Interface (API), nepovolené zásahy do souborů a registrů, porovnávání signatur, vypisování internetového provozu. Podporuje více virtuálních systémů pro analýzu.

Architektura Cuckoo spouští každý vzorek v novém virtuálním prostředí, díky snímkům virtuálních systémů (před každou analýzou automaticky obnovuje dříve vytvořený snímek). Webovým rozhraním je z počítače hostitele nahrán soubor skrze virtuální síť do izolovaných virtuálních prostředí, kde následně proběhne analýza a dalším webovým rozhraním odesílá souhrnné informace k prozkoumání schéma funkce Cuckoo sandbox viz obrázek 3.1. Toto rozhraní využívá M2Crypto jakožto nástroj pro zabezpečení komunikace mezi hostitelem a klienty.



Obr. 3.1: Schéma architektury komunikace Cuckoo [25].

Cuckoo Rooter je funkce umožňující poskytnout virtuálním systémům internetové připojení s několika základními politikami. Přehled politik naleznete v tabulce 3.1. Součástí je podpora InetSim. Pro výpis síťové aktivity Cuckoo používá open sourcové řešení tcpdump. Z hlediska bezpečnosti je samotná instalace Cuckoo sandboxu prováděna pod uživatelem s normálními právy. Je také doporučeno, použít Python Virtual Enviromment (virtualenv). Jedná se o nástroj, který vytvoří izolované prostředí, ve kterém můžeme nainstalovat veškeré Python balíky a spustitelné soubory. Díky tomu se zbavíme jinak potřebných návazností na tyto soubory uvnitř systému hostitele [26]. Navíc není potřeba řešit chybné verze požadovaných balíků pro běh aplikace [27].

Tab. 3.1: Možnosti směrování pomocí funkce Rooter.

None Routing	Žádné směrování paketů. Defaultní nastavení bez využití Rooteru
Drop Routing	Zahazuje veškeré pakety včetně těch určených vnitřní virtuální síti.
Internet Routing	Plný přístup do internetu
InetSim Routing	Směrování do InetSim provozující simulované internetové služby.
Tor Routing	Směrování paketů do sítě Tor
VPN Routing	Směrování skrze VPN

Aktualizování potřebných knihoven třetích stran a samotného Cuckoo je centralizována skrze Cuckoo Working Directory (CWD). Tento adresář sjednocuje všechny potřebné soubory na jednom místě a aktualizaci je možné provést jedním příkazem `pip install -U cuckoo`. Taktéž je možné stáhnout zdroje Cuckoo Comunity, obsahující signatury shrnující chování škodlivých souborů a to za pomocí příkazu `cuckoo community`.

4 Praktická část

Cílem praktické části této semestrální práce je nachystat pracoviště pro následnou instalaci Cuckoo sandbox, správné nastavení programu a provedení analýzy demonstrující možnosti sandboxového prostředí.

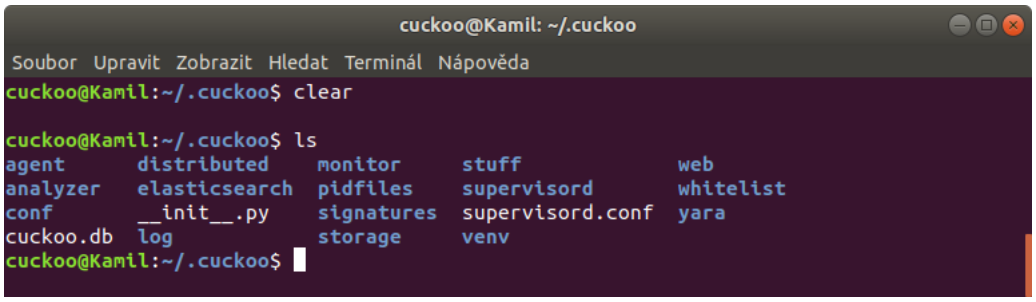
4.1 Příprava

Pro začátek popíši některé důležité části instalace Cuckoo sandbox, se kterými jsem se setkal a mohly by posloužit jako nápověda pro budoucí instalace. Zdůrazňuji, že jako operační systém hostujícího počítače jsem použil Ubuntu 18.04, který je v dokumentaci [28] podrobně popsán a přesto se můžete místy setkat s chybovým hlášením. Veškeré podrobnosti, návod k přípravě a samotné instalace je dostupný v oficiální dokumentaci [28].

První část instalace, dle dokumentace je pojmenována „Requirements“ a shrnuje instalaci knihoven a programů třetích stran s vysvětlením funkce daných komponent. Instalace této sekce se provádí pod uživatelem s *právy správce* (dále jen administrátor), pozdější část instalace bude vyžadovat vytvoření nového uživatele s běžnými právy). Instalace *MongoDB* není nutná, pokud nám k interakci s Cuckoo stačí příkazová řádka, doporučuji však využívat webové rozhraní a tím pádem MongoDB nainstalovat. Jako virtualizační software můžeme zvolit libovolný program, při mém testu jsem použil VirtualBox (verze 5.2.18), který je v dokumentaci označen jako výchozí a jehož použitím se vyhneme nutným modifikacím konfiguračních souborů a instalaci přídavných částí pro podporu funkce jiných virtualizačních softwarů (jako například KVM). Chceme-li ve výsledcích analýzy zkoumat síťovou aktivitu testovaných souborů a webových odkazů, je nutné věnovat pozornost sekci „Installing tcpdump“ části „Requirements“ dokumentace [28]. Zejména, pak pokud používáme výchozí CWD adresář, který je popsán v teoretické části této práce, a jak již bylo zmíněno, usnadňuje orientaci v souborech Cuckoo. Proto věnujte pozornost příkazu `sudo aa-disable /usr/sbin/tcpdump` a zkontrolujte jeho správné provedení příkazem `getcap /usr/sbin/tcpdump`.

Druhá část instalace v dokumentaci [28] nese název „Installing Cuckoo“ a při jejím provedení je třeba dbát na instalaci pod uživatelem s *běžnými právy*, který nemůže využívat nástroj sudo a zasahovat do systému. Tento uživatel se nazývá *standardní uživatel* [29]. O samotné instalaci Cuckoo se následně dočteme, že je vhodné ji provést v již zmíněném virtualenv. Důvody k tomu byly již zmíněny (viz výše), dokumentace [28] však přidává několik dalších. Proto tento krok doporučuji provést a virtualenv prostředí využít. O jeho používání se ještě zmíním dále v práci.

Možnosti konfigurace Cuckoo jsou dosti rozsáhlé, proto zde zmíním alespoň základy nutné pro prvotní testování a spuštění. Cíle využití tohoto sandboxu se mohou lišit a s tím také přichází specifické požadavky na konfiguraci. Nalezení adresáře s konfiguračními soubory je ve výchozím CWD velice snadné pomocí jediného příkazu `cd ~/.cuckoo`. Výpis tohoto adresáře bude podobný jako na obrázku 4.1. Zde vstoupíme do adresáře „conf“ a jako první upravíme soubor s názvem „cuckoo.conf“. Toho můžeme docílit například nástrojem Nano, jehož užití je popsáno v oficiální dokumentaci [30]. Nastavení jednotlivých položek souboru, machinery musí odpovídat zvolenému virtualizačnímu softwaru. Položka „ip“ sekce „resultserver“ nastavuje síť na, které cílové stanice komunikují. Soubor „machinery.conf“ v dokumentaci [28] obecně označuje konkrétní soubor, nesoucí název použitého virtualizačního prostředí, v tomto případě upravuji „virtualbox.conf“. Zde je třeba upravit položku „interface“ , aby souhlasila s virtuální sítí nastavenou pro vnitřní komunikaci virtualboxu. Položka „machines“ definuje názvy dále používané v konfiguračním souboru. V případě více testovacích virtuálních systémů je třeba definovat tyto stroje pod názvem, odlišující se například číslem. Tyto názvy musí být odděleny čárkou. Každý název je poté definován dále v souboru a je nutné nastavit „label“ shodně s názvem systému ve virtuálním softwaru a rovněž definovat „platform“ jako nainstalovaný operační systém a ip adresu v síti definované souborem „cuckoo.conf“.



Obr. 4.1: Výpis Cuckoo adresáře s výchozím umístěním cwd.

4.2 Ukázka analýzy Cuckoo sandbox

Hlavním výstupem této práce je demonstrace funkcí Cuckoo sandboxu. Následující text se věnuje přípravě a uskutečnění analýzy simulovaného útoku, kde na pozici útočníka je virtuální stroj Kali Linux verze 2018.4 a opensourcový nástroj pro penetrační testování Pupy, dostupný na webu pro vývojáře GitHub [31]. Infikovaným počítačem je systém Windows7. Nastavení virtuálních strojů viz tabulka 4.1.

Tab. 4.1: Nastavení jednotlivých operačních systémů.

	Verze systému	CPU (jádra)	RAM	Síťové připojení
Hostitelský PC Ubuntu	Ubuntu 18.04	AMDAthlon II X4 635 (4)	8GB	NAT
VM Windows7	Windows 7 SP1	(2)	2GB	Síť s Hostem (vboxnet0)
VM Kali Linux	Kali Linux 2018.4	(3)	3GB	Síť s Hostem (vboxnet0) / NAT

Kali linux pupy

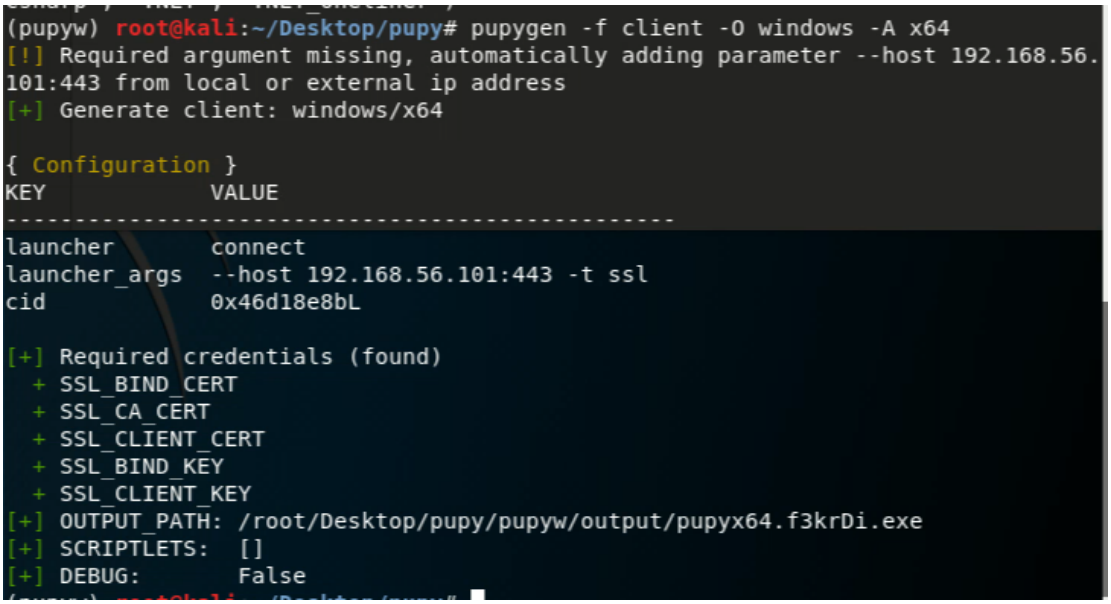
Cílem je vytvořit spustitelný soubor, který bude v rámci analýzy prostřednictvím Cuckoo spuštěn a umožní tak spojení mezi infikovaným počítačem a útočníkem. Vytvoření tohoto vzorku pro následnou simulovanou analýzu je provedeno v prostředí Kali linux, za pomoci multifunkčního nástroje Remote Administration Tool (RAT) generující binární soubory pro vzdálený přístup a následné penetrační testování za pomoci python skriptů [31]. Instalace penetračního nástroje pupy je detailně popsána ne webu GitHub [31]. Prvním krokem k vytvoření testovacího binárního souboru je spuštění virtualenv prostředí příkazem `. activate` v adresáři `pupy/pupyw/bin`. Pomocí python skriptu „pupygen.py“ spuštěného s parametry ve tvaru `pupygen -f client -O windows -A x64` byl vygenerován soubor „pupyx64.f3krDi.exe“ v adresáři `pupy/pupyw/output` viz obrázek 4.2, ze kterého lze vyčíst automaticky vložený parametr `launcher` s hodnotou `connect` a parametry `-host` s automaticky načtenou IP adresou `192.168.56.101` generující Secure Sockets Layer (SSL) spojení, ke kterému se připojí infikovaný počítač. Přepínač `-f` značí formát, parametr `client` značí generování klientského souboru, `-O` definuje cílový operační systém (`windows`) a přepínač `-A` definuje architekturu (`x64` pro 64 bitový systém). Soubor „pupyx64.f3krDi.exe“ naleznete v příloze „**A Obsah přiloženého CD**“.

Spuštění Cuckoo

Cuckoo sandbox je spuštěn opět v prostředí virtualenv a to příkazem `cuckoo` jako na obrázku 4.3. Po zahájení analýzy vzorku toto okno informuje o prováděných akcích viz obrázek 4.4. V chronologickém pořadí tedy Cuckoo připraví virtuální stroj a čeká na vzorek pro analýzu, poté interpretuje parametry nastavené při vložení vzorku a spuštění analýzy. Spustí sniffer pro zaznamenávání síťové komunikace a informuje o průběhu analýzy.

Cuckoo server

Pro komunikaci s Cuckoo jsem zvolil webové rozhraní, proto je nutné pomocí příkazové řádky spustit Cuckoo web server. Prvním krokem je spuštění virtualenv, ná-



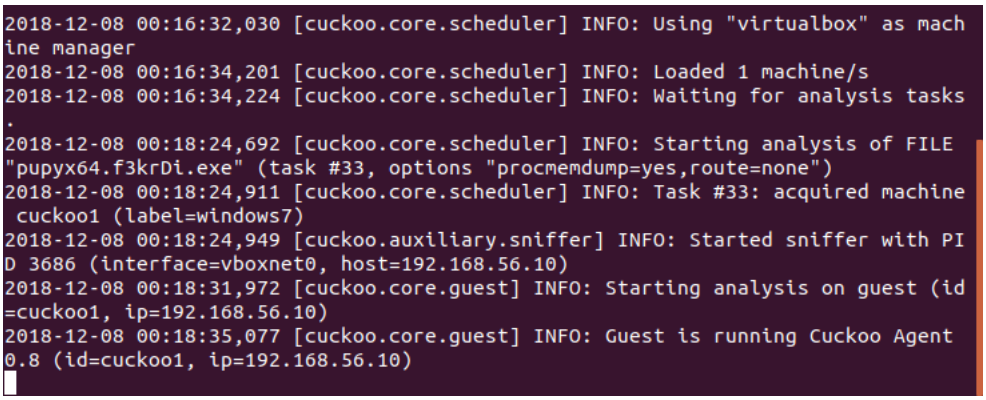
Obr. 4.2: Parametry vygenerovaného binárního souboru.

Obr. 4.3: Hlášení při úspěšném spuštění Cuckoo.

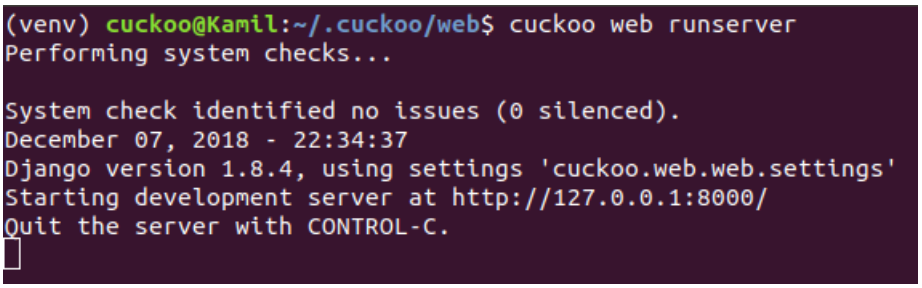
sledně se příkazem `cd ~/.cuckoo/web` přepneme do CWD a zadáním `cuckoo web runserver` spustíme server viz obrázek 4.5. Cuckoo web pracuje skrze Django aplikaci a ve výchozím stavu používá adresu `127.0.0.1:8000`.

Cuckoo webové rozhraní

Pro otevření rozhraní je nutné zadat předem definovanou adresu a port z konfiguračního souboru do libovolného prohlížeče, v tomto případě Mozila firefox 63.0.3(64bit),



Obr. 4.4: Výpis Cuckoo terminálu při spuštěné analýze.



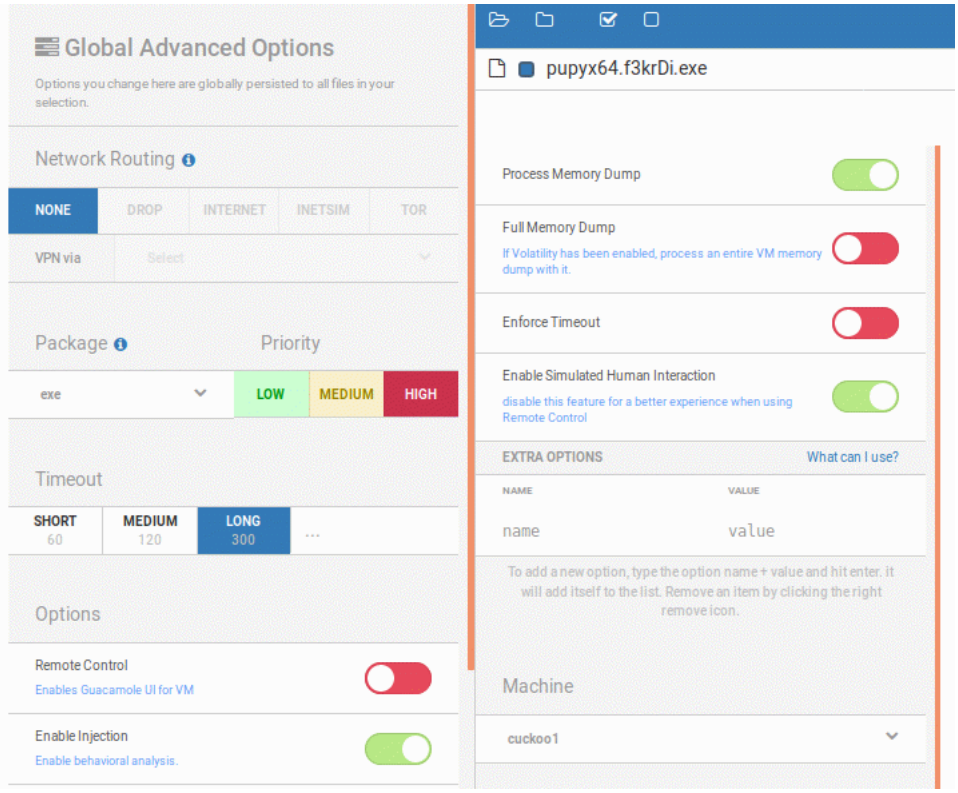
Obr. 4.5: Cuckoo web server po spuštění v terminálu.

dostupného z webu Mozila.cz [32]. Prohlížeč je nyní schopen skrze Localhost Loopback Addresses získat data ze spuštěného Cuckoo web serveru. Jedná se o adresu lokální sítě daného zařízení (Ubuntu)[33].

Nastavení analýzy

Ve webovém rozhraní je možno pomocí GUI nahrát soubor k analýze pouhým přetažením z adresářové struktury. Následným potvrzením spuštění se dostáváme k samotnému nastavení viz obrázek 4.6. Položka „Network Routing“ je nastavena na hodnotu „NONE“, jelikož pro komunikaci není třeba žádného přístupu na internet a veškerá komunikace běží lokálně. „Package“ s nastavením „exe“ definuje některé speciální postupy při analýze spustitelných souborů[28]. Pole „Timeout“ s hodnotou „300“ značí dobu trvání analýzy v sekundách. Přepínač „Remote Control“ je vypnutý, jelikož není třeba testovací stroj vzdáleně ovládat, „Enable injection“ povolen pro analýzu chování vzorku, „Process Memory Dump“ rovněž povolen. Položka „Full Memory Dump“ je zakázána z důvodu nevyužití statické analýzy, „enforce Timeout“ rovněž není povolena pro případ, že kroky na stanici Kali linux potrvají

déle než doba trvání nastavená na začátku a neprovedení všech útoků. Pole „Enable Human Interaction“ je povoleno pro umělé napodobení chování uživatele.



Obr. 4.6: Možnosti nastavení analýzy vzorku.

Spuštění analýzy

Po nastavení parametrů je za pomocí GUI webového rozhraní možné spustit analýzu. Průběh v terminálu je možné vidět na obrázku 4.4. V tomto kroku bylo zapotřebí spustit virtuální stroj Kali linux a spouštět jednotlivé moduly pro vyzkoušení, jaké informace je možné následně z výsledků analýzy Cuckoo získat. Spojení mezi Kali a virtuálním strojem řízeným Cuckoo je navázáno ihned po spuštění a je zobrazeno v aplikace pupysh.

Kroky v Kali linux

Pro spuštění použijeme virtualenv prostředí stejně jako při generování spustitelného souboru příkazem `. activate` v adresáři `pupy/pupyw/bin`. Komunikace s klientem (virtuálním strojem Windows 7) je zprostředkována python skriptem spuštěným příkazem `pupysh`, kde přikazem `--help -M` můžeme zobrazit veškeré moduly a příkazy pro penetrační testování. V rámci tohoto testu jsem použil několik příkazů a

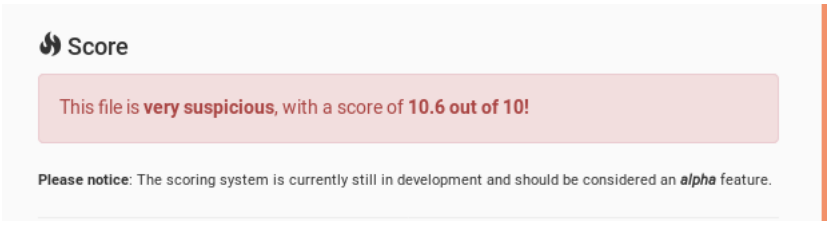
modulů jež spolu s vysvětlením naleznete v tabulce 4.2. Vzhledem k povaze těchto modulů a jejich funkci bude ve výstupu pátráno po detekci zásahů do systému a síťové komunikaci.

Tab. 4.2: Přehled využitých příkazů a modulů v pupysh.

Příkazy	**Vysvětlení**
sessions	Výpis navázaných spojení
run	Spuštění testovacích modulů
Moduly	**Vysvětlení**
Gather/keylogger	Zachytávání klientovy klávesnice
Gather/Mouselogger	Zachytávání klientova kliknutí se screenem
Manage/Download	Stažení souboru z klientova počítače
Network/port_scan	Skenování otevřených portů sítě
Troll/msgbox	Výpis zprávy na klientovu obrazovku

4.3 Procházení výstupu analýzy

Po skončení analýzy, které je determinováno vypršením limitu nebo ztracením spojení se vzdáleným SSL spojením, přichází na řadu procházení výstupu Cuckoo analýzy. Jako první informaci nám webové rozhraní poskytne souhrn zjištění analýzy. První indikátor, zda je testovaný soubor škodlivý či nikoliv je položka Score viz obrázek 4.7. Toto skóre i přes varování, že se jedná o prvek v testování, dobře vypovídá o dopadu testovaného souboru na systém. Před testováním tohoto binárního souboru, jsem provedl několik předběžných analýz různých vzorků a modulů pro penetrační testování, s cílem určit, jak moc skóre vypovídá o nebezpečí vzorku. Po provedení testu cirka třiceti vzorků mohu konstatovat, že hodnoty vyšší než 3 již vypovídají o nebezpečném chování vzorku a je vhodné se jimi dále zabývat. V tomto případě bylo dosaženo skóre 10.6 z 10.



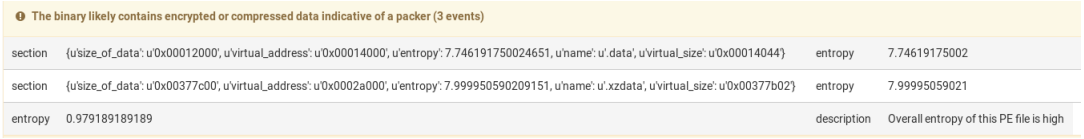
Obr. 4.7: Skóre hodnocení nebezpečnosti vzorku.

Rozpoznání škodlivého souboru

Potenciální hrozbu ze strany testovaného souboru lze zjistit již při analýze samotného vzorku, bez interakce s Kali linux a to na základě výsledků analýzy na obrázku 4.8. Vysoká míra entropie vyvolává podezření, že se může jednat o tzv. „Packer“ , nebo „Crypter“. Cuckoo analyzovaný soubor rozdělí na sekce a ty následně testuje za pomoci následující Shannonovy rovnice:

$$H = -\sum_{i=0}^{255} P_i log_2(P_i) \tag{4.1}$$

Výsledkem je číslo mezi 0 a 8, čím blíže je toto číslo osmi tím víc náhodný je obsah sekce. V případě této analýzy můžeme vidět hodnotu 7.746 a 7.999. Cuckoo na závěr vyhodnotí celkovou entropii souboru spolu s hodnotou vyjádřující poměř komprimovaných dat ku celkové velikosti souboru, v tomto případě 0.979. Tuto funkci má na starosti python modul `packer_entropy.py` [34]. Účel detekce vysoké entropie spustitelného souboru, je upozornit na potenciální maskování vnitřních řetězců a vzorců, které by bez šifrování a zabalení byly ihned odhaleny [35].



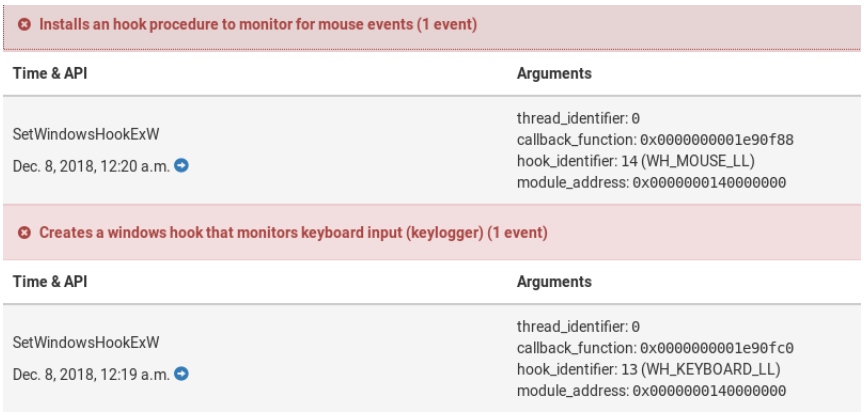
Obr. 4.8: Hlášení o vysoké míře entropie vzorku.

Detekce vysoké entropie je následně podpořena detekcí alokované paměti pro rozbalení sama sebe viz obrázek 4.9. Nutno dodat, že tato signatura je běžná při analýze souborů a slouží jako informativní doplnění jiných signatur.

Obr. 4.9: Alokování paměti pro rozbalení škodlivého obsahu.

Detekce sledování uživatele

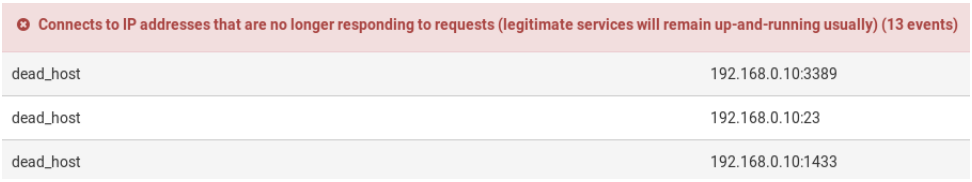
Dva moduly spuštěné v pupy, byly určeny k monitorování uživatele, keylogger a mouslogger viz tabulka 4.2. Cucukoo tuto aktivitu automaticky detekuje a informuje o ní v přehledu analýzy viz obrázek 4.10.



Obr. 4.10: Detekce vytvoření a instalace sledování myši a klávesnice.

Síťová komunikace

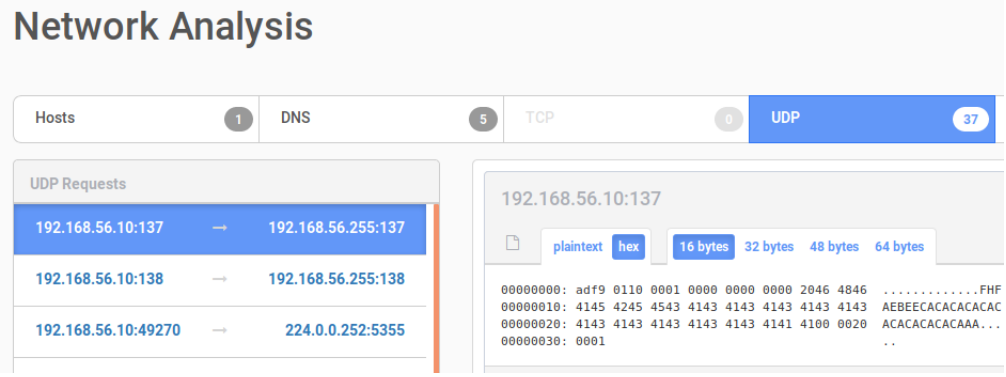
V přehledu se podařilo objevit skenování portů provedené v pupy viz obrázek 4.11. Zde jsou vidět pokusy o komunikaci s dále neodpovídajícími porty 23, 1433 a 3389 na IP adrese 192.168.0.10, která byla zvolena jako testovací podsíť na straně útočníka. Analýze chování odhalila komunikaci s adresou 192.168.56.101 na portu 433 viz obrázek 4.13. Tato adresa slouží k připojení napadeného systému k útočníkovi. Pro podrobnější informace o proběhlé komunikaci, byla využita analýza sítě s názvem „Network“ viz obrázek 4.12. Z ní však není patrná žádná podezřelá komunikace a detekuje pouze User Datagram Protocol (UDP) provoz. Bližší průzkum ukazuje, že hlavní komunikace na portu UDP směřuje na všesměrovou adresu NetBIOS [36].



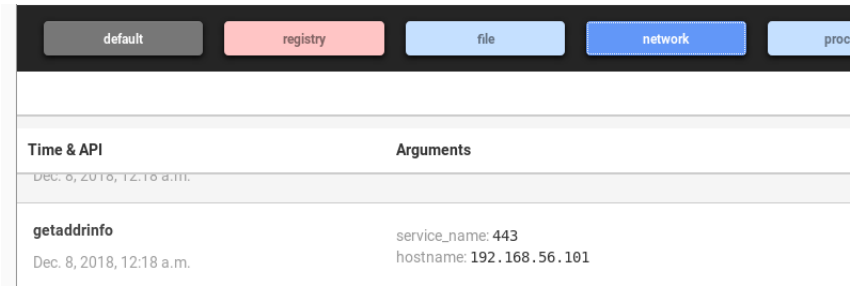
Obr. 4.11: Detekce skenování portů.

Změny v registrech

Při použití modulu privesc viz tabulka 4.2, je zapotřebí provést Dynamic-link library (DLL) injection. Průběh spuštění tohoto modulu najdeme na obrázku 4.14. Tento krok je na straně Cuckoo detekován v „Summary“ a to hláškou o přebrání práv pro spuštění jiného procesu s poznámkou, že se může jednat o vložení kódu (což v tomto případě opravdu je).



Obr. 4.12: Analýza sítě protokolu UDP.



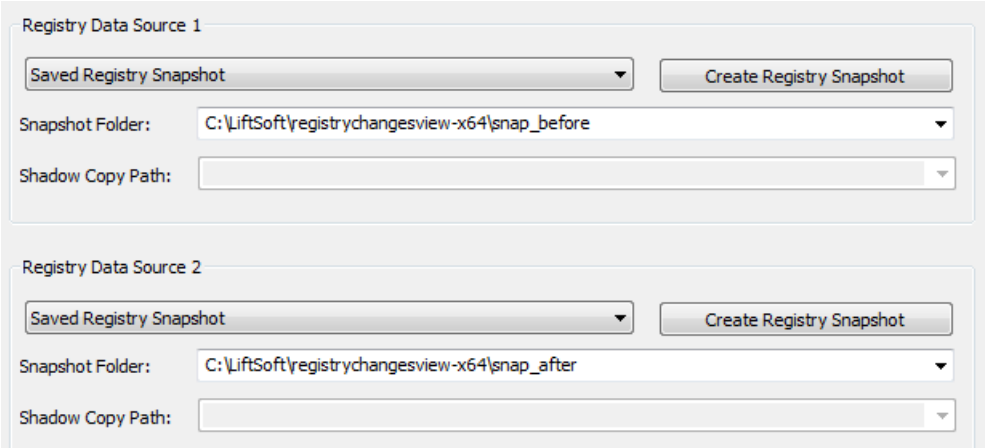
Obr. 4.13: Úspěšné zjištění IP adresy útočníka.

Jednotlivé exploitační příkazy nástorje pupy, byly provedeny také zvlášť s využitím programu RegistryChangesView dostupného na stránkách `www.nirsoft.net` [37]. Díky němu lze porovnávat registry před a po provedení změny viz obrázek 4.15. Na základě této techniky srovnání bylo zjištěno, že pupy využívá především registry na obrázku 4.16.Ty byly pro ukončení procesu upraveny nebo odstraněny. Ostatní exploitační techniky nezanechali žádnou změnu v registrech. Jedinou výjimkou byl nástroj keylogger, jímž změněné registry jsou na obrázku 4.17.

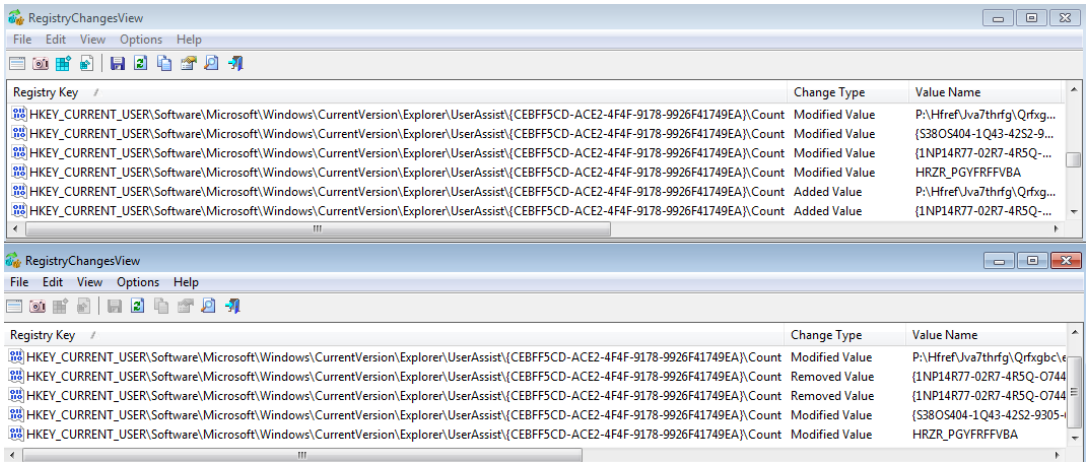
```
KEY             VALUE
-------------------------------------------------
launcher        connect
launcher_args   --host 192.168.56.101:443 -t ssl
cid             1188138635

[+] Required credentials (found)
  + SSL_BIND_CERT
  + SSL_CA_CERT
  + SSL_CLIENT_CERT
  + SSL_BIND_KEY
  + SSL_CLIENT_KEY
[+] Template: pupyx64.dll
[+] injecting DLL in target process 2244 ...
```

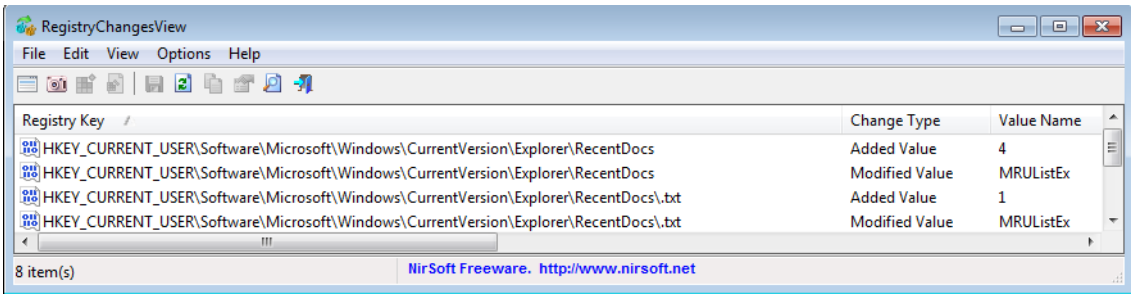
Obr. 4.14: Proces získání systémových práv v Pupy.



Obr. 4.15: Srovnání registrů před a po provedení určitého útoku.



Obr. 4.16: Hlavní registry využité programem pupy.



Obr. 4.17: Registry změněné modulem keylogger.

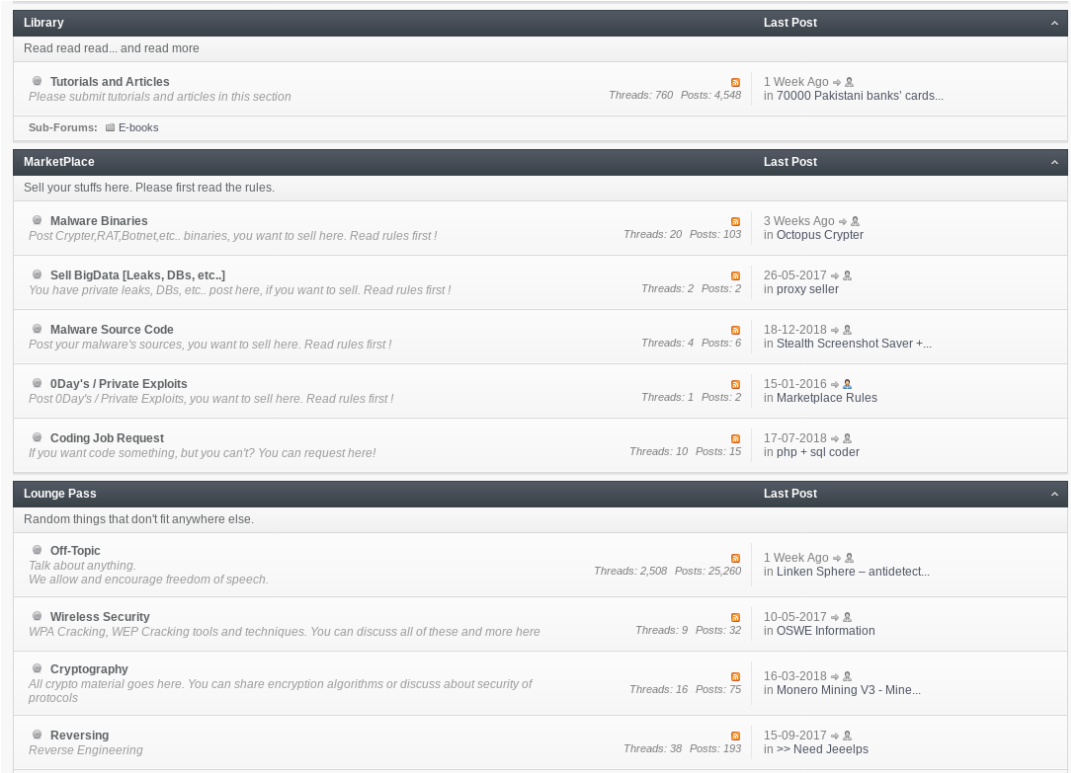
4.4 Analýza webu se zaměřením na malvertising

Tato praktická část bakalářské práce se věnuje dynamické analýze potenciálně nebezpečných webových stránek se zaměřením na odhalení malvertizing kampaně. Procesu analyzování webového prostoru pomocí sandboxu předcházelo prozkoumání prostředí výskytu škodlivého kódu, kterým se budu věnovat v úvod této části bakalářské práce.

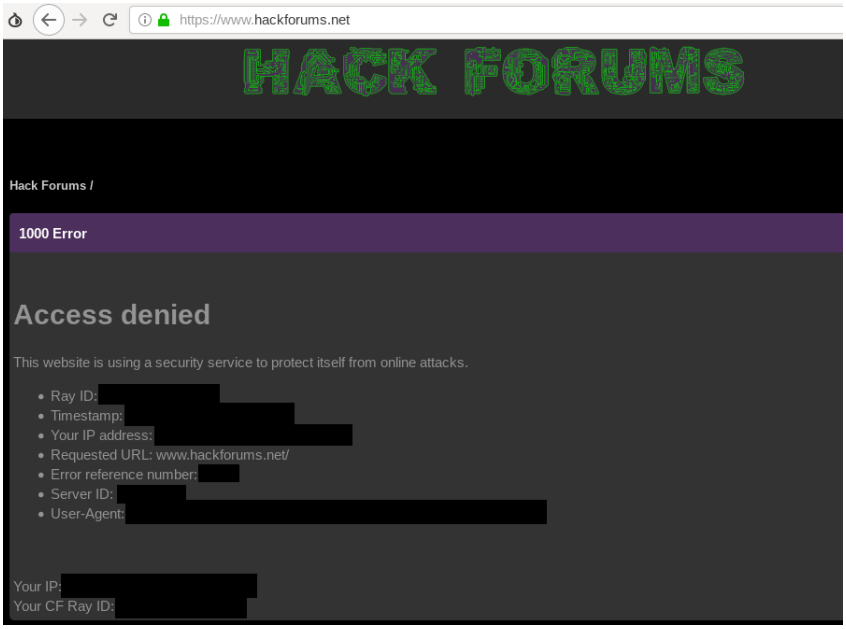
Cílem této části práce bylo otestování Cuckoo sandboxu v reálném prostředí, s možným odhalením volně šířeného škodlivého kódu v internetu a to dynamickou analýzou webových stránek. Dále najít zajímavé vzorky škodlivých kódů, analyzovat je dynamicky i staticky a sdílet odhalené vzorky na příslušných webech zabývajících se kybernetickou bezpečností.

4.4.1 Underground fóra pro šíření malwaru

Abych získal na problematiku náhled z druhé strany, navštívil jsem některá fóra zabývající se tématikou malwaru a hackingu. Zde jsou celé sekce a diskuze věnující se tématu malwaru, spolu s návody, vzorky a výukovými kurzy. Je možné najít také obchody nabízející zero day vzorky, sloužící například jako finální payload v rámci malvertizingu. Pro ilustraci uvádím snímek obrazovky na obrázku 4.18. Jedná se o lehce přístupné fórum a osobně bych ho popsal jako edukační, přesto je zde například sekce `MarketPlace` pro sdílení malwaru. Pokročilejší fóra nejsou otevřená všem a je nutné získat si nějakým způsobem důvěru komunity. Další vyžadují vstupní poplatky nebo přijímá uživatele na základě předem definovaných pravidel. Příklad fóra s omezeným přístupem je na obrázku 4.19. Je důležité zmínit, že pro procházení takovýchto fór je vhodné využít Tor Browser, na některé se bez něj nelze ani dostat. Je možné také využít VPN ve spojení s Tor Browserem pro zajištění anonymity již při vstupu na síť Tor. Instalace prohlížeče je možná například pomocí příkazu Advanced Package Tool (APT) a to v linuxových operačních systémech na bázi Debianu [38]. Příkaz pro instalaci je ve tvaru `$ sudo apt install torbrowser-launcher`.



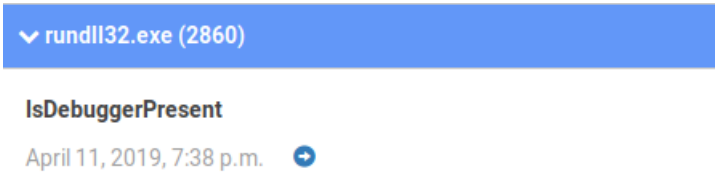
Obr. 4.18: Náhled sekcí fóra [39].



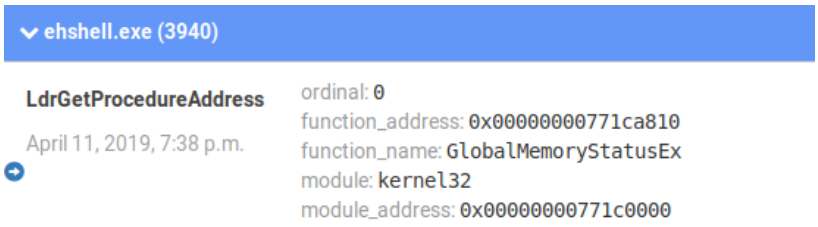
Obr. 4.19: Zamítnutí přístupu na fórum [40].

4.4.2 Analýza na systému Windows 7

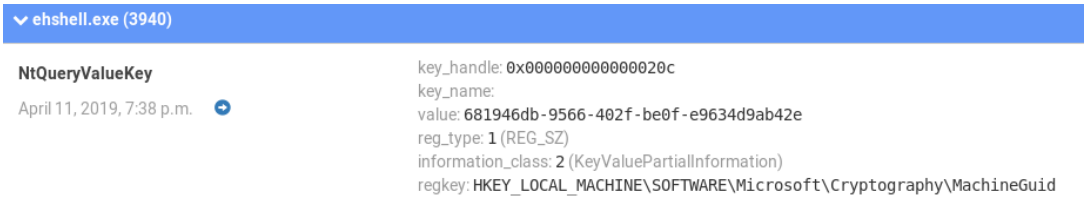
Pro analýzu webu jsem zvolil již využitý systém Windows 7 při simulované analýze popsané v části práce **Ukázka analýzy Cuckoo sandbox** 4.2. Prostředí bylo doplněno o funkci VPN Routing z dostupných možností součásti Cuckoo sanboxu „Cuckoo rooter“. Vzhledem k vývoji hledání malvertizing kampaní byl virtuální systém upraven pro ztížení odhalení virtuálního prostředí malwarem, zejména pak upravením hardwarových specifikací a odstraněním problémů spojených s detekcí virtuálního prostředí za pomoci API volání a analýzy chování Cuckoo sandbox viz obrázky 4.20 4.21 4.22 4.23.



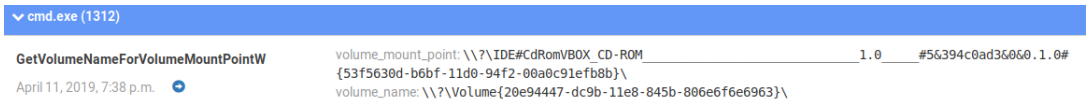
Obr. 4.20: Detekce API volání pro zjištění přítomnosti debuggeru.



Obr. 4.21: Detekce API volání pro zjištění velikosti přidělené operační paměti.



Obr. 4.22: Detekce API volání pro zjištění unikátního identifikátoru systému.



Obr. 4.23: Detekce API volání pro zjištění připojovacího bodu.

Nalezení vhodné webové stránky

Pro začátek bylo třeba zajistit vhodný soubor potenciálně nebezpečných stránek, na kterém je možné selektovat specifické prvky chování z výsledků prvotních sandboxových analýz. Jako vzorek jsem vybral sto webových odkazů zachycených firewallem a zahájil první sadu testování viz obrázek 4.24. Prvotním identifikátorem pro následnou podrobnější analýzu je hodnocení udělené Cuckoo. V rámci takto analyzovaných vzorků jsem ověřoval problémy stránek vyhodnocené Cuckoo sandboxem jako podezřelé, tak abych vybral vzorek s opravdu podezřelým chováním k následné hlubší analýze. Jako první jsem provedl síťovou analýzu na základě hlášení o komunikaci se stranou bez využití Domain Name System (DNS) viz obrázek 4.25.

Díky této analýze se mi podařilo najít opakující se IP adresy, nebo alespoň podsítě, se kterými stránky komunikují. Při důkladnějším prozkoumání těchto adres pomocí serveru DNSQUERY jsem mohl pokračovat v analýze stránek odkazující se ke stejným serverům [41]. Tato síťová analýza je pouze doplňující a vzhledem k původu koncových serverů na ní nebylo mým cílem navazovat dalším zkoumáním. Výsledkem tedy bylo nalezení často se opakujících serverů, například:

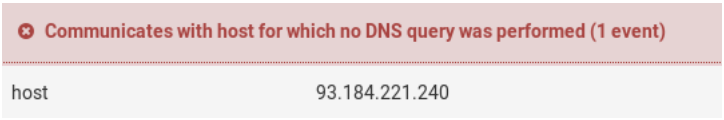
- IP(93.184.220.29) - vlastník: Derrick Sawyer, málo informací zjištěných za pomoci DNSQUERY.
- IP(95.100.184.155) - vlastník: AKAMAI, firma poskytující mimo jiné cloudové služby.
- IP(197.156.74.193) - vlastník: ethio telecom, poskytovatel síťových služeb.

4.4.3 Hlubší analýza vybrané stránky

Nejvyššího hodnocení dosáhla stránka `http://zdajecia.nurka.pl` konkrétně 5.2. Vzhledem k dlouhým názvům vzorků a odkazů nalezených při analýzách, budu používat upravené názvy pro jejich označení v textu a původní jména spolu s dodatečnými informacemi uvedu v patřičných tabulkách. Důležité soubory naleznete v příloze „**A Obsah přiloženého CD**“. Vybranou webovou stránku budu dále označovat jako ***Odkaz1*** viz tabluka 4.3. Analýza odhalila vysoký počet signatur poukazujících na nebezpečné chování stránky viz obrázek 4.26. Nyní se zaměřím na rozbor jednotlivých signatur a analýzu některých z nich.

107	2019-03-31 12:20	-	http://bestpage.cz	reported	score: 1.8
106	2019-03-26 14:01	-	http://znackova-obuv.cz	reported	score: 3.6
105	2019-03-26 13:59	-	http://zdravavyziva.skherbalife.sk	reported	score: 3.6
104	2019-03-26 13:57	-	http://zdjecia.nurka.pl	reported	score: 5.2
103	2019-03-26 13:54	-	http://zapper.cz	reported	score: 4.8
102	2019-03-26 13:52	-	http://yessport.slaskdatacenter.pl	reported	score: 2.8
101	2019-03-26 13:51	-	http://xoomer.virgilio.it	reported	score: 4
100	2019-03-26 13:49	-	http://xhr.yallboen.com	reported	score: 2.8
99	2019-03-26 13:47	-	http://www.zoyaweddingcenter.sk	reported	score: 2.8

Obr. 4.24: Výběr webových stránek z úvodního souboru odkazů.



Obr. 4.25: Signatura poukazující na komunikaci bez využití DNS.

Tab. 4.3: Tabulka identifikace prvotně analyzovaného webového odkazu.

Název v textu	Odkaz1
Webový odkaz	http://zdjecia.nurka.pl
Číslo analýzy souboru	104

HTTP požadavky (Performs some HTTP requests)

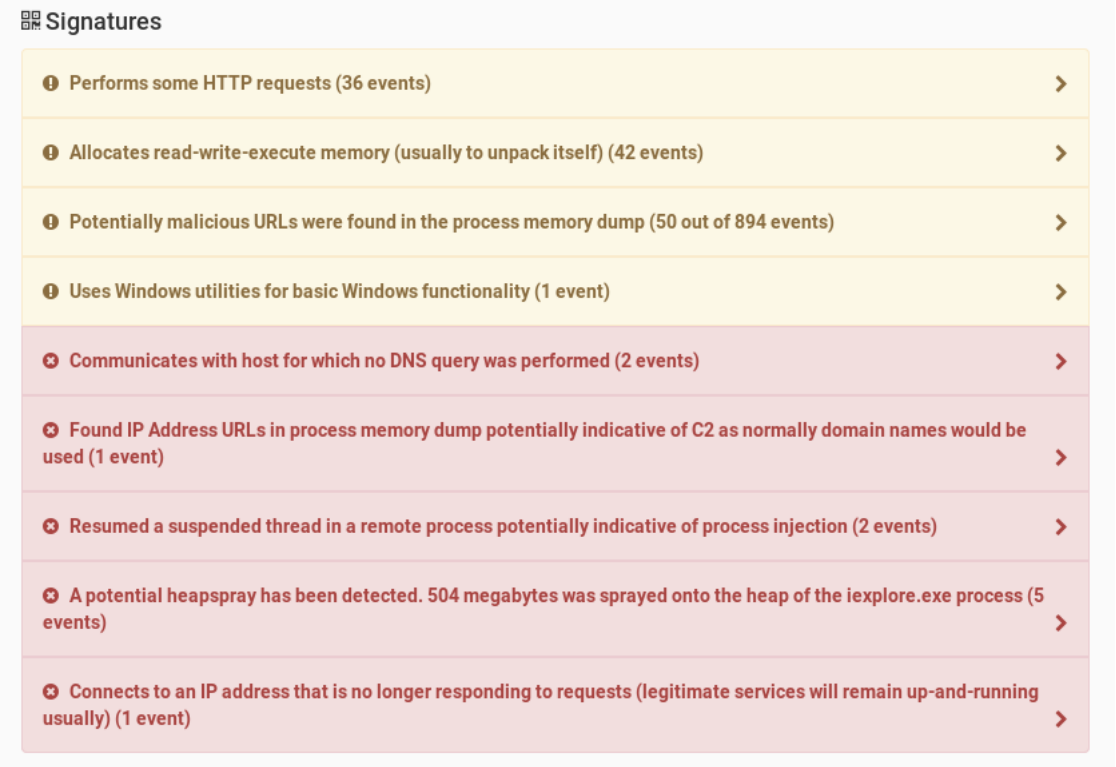
Vysoký počet těchto požadavků nebývá obvyklý a v některých případech je zde možné nalézt odkazy k prozkoumání, vedoucí k novým poznatkům. V tomto případě jsem k žádným nedospěl.

Vyčlenění paměti pro rozbalení (Allocates read-write-execute memory)

Tato signatura je přítomna u většiny analýz a pro mé účely nebyla podstatná.

Potenciálně nebezpečné odkazy (Potentially malicious URLs were found)

Ani v tomto případě nejde o podstatnou signaturu. Jedná se o webové odkazy v paměti procesu a opět nepřináší mnoho cenných informací.



Obr. 4.26: Výpis signatur nalezených pomocí Cuckoo.

Využití nástrojů Windows (Uses Widnows utilities)

Zde je možné nalézt informace o využití nástrojů, které by využívány být neměly. V rámci mé analýzy webových odkazů tato signatura neměla zásadní přínos.

Komunikace s třetí stranou bez využití DNS(Communicates with host for which no DNS query was performed)

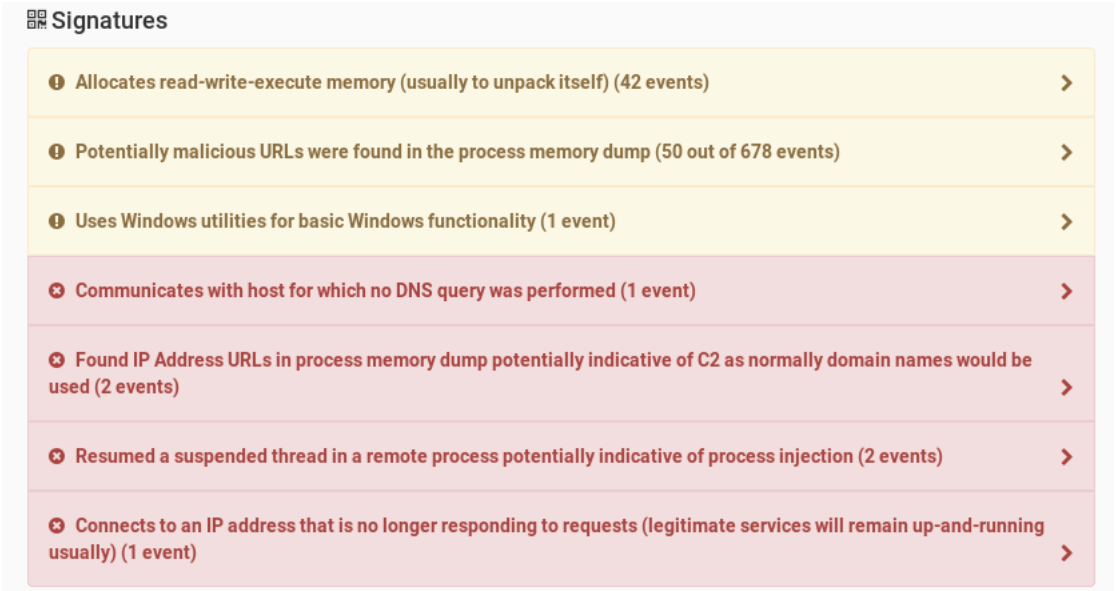
Tato signatura je detailněji popsána výše. V případě tohoto vzorku je komunikováno s adresou 67.27.151.254 (Level 3 Parent,LLC poskytovatel telekomunikačních služeb) a 81.95.96.126 (active24.cz poskytovatel webových služeb)

Nalezení IP adresy v paměti na místo doménového odkazu(Found IP Address URLs in process memory dump potentially indicative of C2)

Položka této signatury značí neobvyklé chování stránky opuštěním původní domény. Nalezený odkaz je proto vhodný k dalšímu zkoumání. V rámci analýzy jsem našel ***Odkaz2*** viz tabulka 4.4. Po jeho samostatné analýze bylo zjištěno několik signatur viz obrázek 4.27.

Tab. 4.4: Tabulka identifikace nalezeného odkazu využívajícího IP adresu.

Název v textu	Odkaz2
Webový odkaz	http://74.50.124.201/image/images2_full/XBdG-1ls-1_qzRhbjcSua.gif
Odkaz získán v analýze číslo	104
Číslo analýzy souboru	114



Obr. 4.27: Výpis signatur analyzovaného odkazu2.

Na základě počtu a typu signatur spolu s vysokým skóre 4.0, jsem v rámci analýzy tohoto vzorku prozkoumal další z podrobných prvků Cuckoo zahrnujících analýzu chování, stažené soubory a síťovou analýzu pcap. souboru pomocí programu Wireshark. V tomto případě však nevedly k nalezení zásadních informací a na základě podobnosti signatur s Uniform Resource Locator (URL) ***Odkaz1*** jsem se na ***Odkaz2*** dále nezaměřoval.

Potenciální injekce procesu (Resumed a suspended thread in a remote process)

Tato signatura varuje před možnou injekcí procesu. Tento incident by byl jasnou ukázkou škodlivého chování, bohužel v rámci všech analýz byla tato signatura detekována a to i na bezpečných serverech. Proto jsem tuto signaturu nebral jako spolehlivý indikátor.

Potenciální heapspray (A potential heapspray had been detected.)

Indikace heapspray útoku je ukazatelem, že webový odkaz nepracuje správně, nebo se snaží o vložení souborů potřebných pro exploit do paměti heap. V každém případě

je nutné tuto signaturu prověřit. Vzhledem k falešné indikaci způsobené dlouhými datovými toky, například multimediálním streamem, jsem využil extrahovaného .pcap souboru. Po jeho otevření ve Wiresharku, se mi nepodařilo odhalit chybu v síťovém protokolu, který by zapříčinil falešnou indikaci heapspray útoku. Pokusil jsem se tedy o analýzu memory dumpu pomocí olydbg a x64dbg [42] [43]. Nebylo však možné tento dump spustit a tím se na heapspray podívat blíže. Nemohu tedy potvrdit ani vyvrátit předpoklad o útoku, v každém případě samotný výpis Cuckoo popisuje rozsah o velikosti 504 megabajtů. Vzhledem ke zjištění v následujících analýzách se přikláním k variantě heapspray útoku než-li k falešnému hlášení.

Připojení k adrese která nadále odpovídá (Connects to an IP address that is no longer responding)

Toto hlášení může mít několik příčin. Cílový server může být náhle odpojen nebo zablokován. Adresa může patřit vzdálenému serveru připravenému na provedení škodlivých příkazů či výměnu souborů pro další exploitační techniky. Komunikace na tomto portu byla pouze obslužná a selhala při pokusu o resetování spojení, což vyplývá z extrahovaného pcap. souboru.

4.4.4 Stažené soubory

Tato sekce Cuckoo sandboxu shromažďuje soubory, které byly staženy v rámci analýzy vzorku. Při analýze vybrané stránky bylo nalezeno 42 takovýchto souborů. Existuje vícero způsobů jak s touto sekcí naložit a získat tak nové informace k celkové analýze. Je možné se zaměřit pouze na určitý typ souborů, který lze následně stáhnout a podrobit nové analýze viz obrázek 4.28. Možností je například zamě-

Name	**6cf156d528af2d17_gpt[1].js** Download Submit file
Filepath	C:\Users\Win7guest\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.IE5\4WVMYB37\gpt[1].js
Size	42.9KB
Processes	2736 (**iexplore.exe**)
Type	ASCII text, with very long lines
MD5	e8eaab0fd0521b75e78d60ea5fad5a3d
SHA1	1ad81310e9aea126328b7997e9907233ccce7af8
SHA256	6cf156d528af2d1791543fa0eddc3e1cef2f522a79bdc8ce47ba99440755a1eb
CRC32	5B731A52
ssdeep	None
Yara	None matched
VirusTotal	Search for analysis

Obr. 4.28: Ukázka staženého java scriptu.

ření na obrazové soubory typu .gif, .png, atd. a zaměřit se tak na oblast šifrování dat do grafických souborů. Já jsem se zaměřil na soubory JavaScriptu vzhledem k

jejich dobré využitelnosti v malvertizing kampaních a obecně častému použití při expoitaci. Analýza vybrané stránky odhalila celkem 6 souborů javaScriptu. Tyto soubory jsem jednotlivě analyzoval Cuckoo sandboxem, avšak bez nalezení podstatných signatur. Přistoupil jsem tedy ke statické analýze, za pomoci textového editoru jsem procházel scripty ve snaze najít jakékoli indikátory škodlivého záměru. V případě souboru `6cf156d528af2d17_gpt[1].js`, v práci označená jako ***JavaScript1*** viz tabulka 4.5 se jednalo o záměrně obfuskovaný kód, což je možné vidět na obrázku 4.29. Díky stránce deobfuskující takovýto kód, se mi podařilo nalézt schovaný webový odkaz viz obrázek 4.30

Tab. 4.5: Tabulka identifikace nalezeného JavaScriptu.

Název v textu	JavaScript1
Název souboru	6cf156d528af2d17_gpt[1].js
SHA256 Hash	6cf156d528af2d1791543fa0eddc3e1cef2f522a79bdc8ce47ba99440755a1eb
Soubor získán v analýze číslo	104
Číslo analýzy souboru	129

```
(function(F,S){var window=this;var aa="function"==typeof Objec
b=function(){};b.prototype=a;return new b},ba;if("function"==t
Object.setPrototypeOf)ba=Object.setPrototypeOf;else{var ca;a:{
0},ea={};try{ea.__proto__=da;ca=ea.H;break a}catch(a){}ca=!1}b
{a.__proto__=b;if(a.__proto__!==b)throw new TypeError(a+" is n
fa=ba,ha=function(a,b){a.prototype=aa(b.prototype);a.prototype
for(var c in b)if("prototype"!=c)if(Object.defineProperties){v
d=Object.getOwnPropertyDescriptor(b,c);d&&Object.defineProper
a[c]=b[c];a.I=b.prototype},g=this,k=function(a){return"string"
{return"number"==typeof a},ja=function(){if(null===n)a:{var
a=g.document;if((a=a.querySelector&&a.querySelector("script[no
a.getAttribute("nonce"))&&ia.test(a)){n=a;break a}n=""}return
```

Obr. 4.29: Ukázka záměrně obfuskovaného kódu java script.

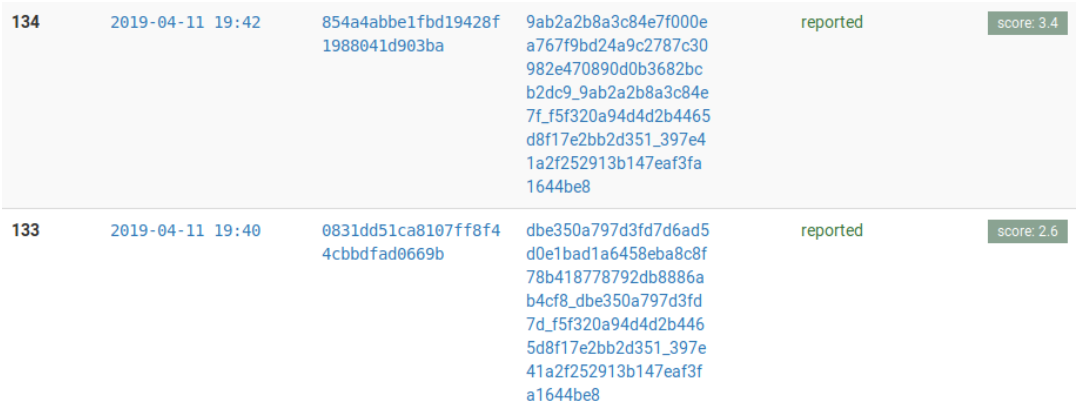
```
2856        ]],
2857        [100, [
2858            [21063348, [
2859                [null, null, 9, [null, null, "https://securepubads.g.doubleclick.net/pagead/js/rum.js"]]
2860            ]]
2861        ]],
2862        [1, [
2863            [21063363],
2864            [21063364, [
2865                [null, 16, null, [null, 500]]
2866            ]],
2867            [21063365, [
2868                [null, 16, null, [null, 1000]]
```

Obr. 4.30: Nalezený odkaz v deobfuskovaném kódu [44].

4.4.5 Analýza nalezeného URL Odkaz2

Ve výsledcích URL jsem nenašel nikterak závažné indikace nebezpečného chování. Jedna z nalezených signatur poukazovala na změnu práv ze zápisu a čtení na čtení

a spuštění. Pro získání zajímavých poznatků jsem opět využil stažených souborů. Celkem odkaz stáhl pět souborů, které jsem podrobněji zkoumal. Jako zajímavé se ukázaly dva binární soubory viz obrázek 4.31. Tyto soubory budu v práci označovat jako ***Dropper1*** viz tabulka 4.6 a ***Dropper2*** tabulka 4.7 vzhledem k signaturám poukazující na tento druh malwaru. Podrobný popis analýz je sepsán níže.



Obr. 4.31: Binární soubory detekované jako droppery.

Tab. 4.6: Tabulka identifikace souboru v textu označeném jako Dropper1.

Název v textu	Dropper1
Název souboru	dbe350a797d3fd7d6ad5d0e1bad1a6458eba8c8f78b418778792db8886ab4cf8 _dbe350a797d3fd7d_f5f320a94d4d2b4465d8f17e2bb2d351_397e41a2f252913b147eaf3fa1644be8
SHA256 Hash	dbe350a797d3fd7d6ad5d0e1bad1a6458eba8c8f78b418778792db8886ab4cf8
Soubor získán v analýze číslo	134
Číslo analýzy souboru	133

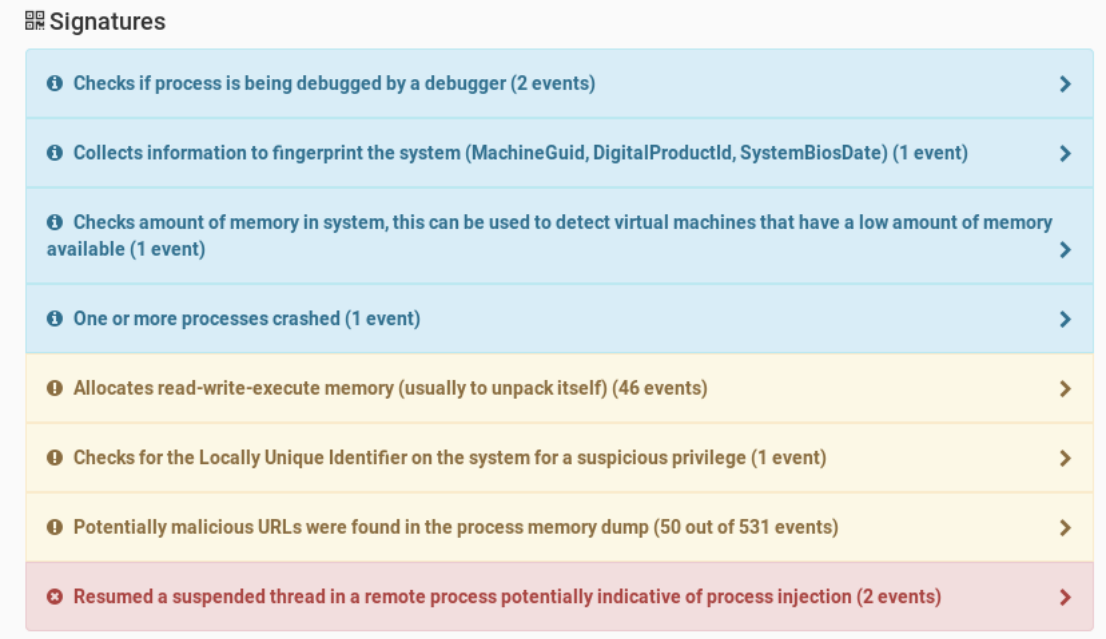
Tab. 4.7: Tabulka Identifikace souboru v textu označeném jako Dropper2.

Název v textu	Dropper2
Název souboru	9ab2a2b8a3c84e7f000ea767f9bd24a9c2787c30982e470890d0b3682bcb2dc9 _9ab2a2b8a3c84e7f_f5f320a94d4d2b4465d8f17e2bb2d351_397e41a2f252913b147eaf3fa1644be8
SHA256 Hash	9ab2a2b8a3c84e7f000ea767f9bd24a9c2787c30982e470890d0b3682bcb2dc9
Soubor získán v analýze číslo	132
Číslo analýzy souboru	134

Dynamická analýza souboru Dropper1

Výčet rozebíraných signatur je možné vidět na obrázku 4.32. První signatura poukazuje na dvojí volání funkce `IsDebuggerPresent`. V obou případech je návratová

hodnota `0`, což dle oficiální dokumentace microsoft znamená, že proces není debugován [45]. ***Dropper1*** voláním této funkce zjišťuje, zda má být spuštěný celý program a používá toto volání jako proti detekční opatření. Druhá signatur čte registr `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\MachineGuid` pro definování jedinečného identifikátoru systému. Opět se jedná o proti detekční opatření. Třetí signatura používá funkci `GlobalMemoryStatusEx`, další technika pro detekci sandboxu a určení velikosti virtuální a fyzické paměti. Čtvrtá signatura poukazuje na pád procesu, v tomto případě sekundárního `rundll32.exe` spuštěného pod procesem `cmd.exe`. Pro detailnější pohled na tuto signaturu je možné využít analýzu chování viz obrázek 4.33, z níž se dozvídáme o spuštění celkem tří procesů s níž souvisí osmá signatura o potenciální injekci procesu viz obrázek 4.34. Na závěr šestá signatura ukazuje volání funkce `LookupPrivilegeValue` s hodnotou `SeDebugPriviledge`, která povoluje převzetí kontroly nad nesystémovým procesem a umožní tak na něm volajícímu procesu využít windows API.

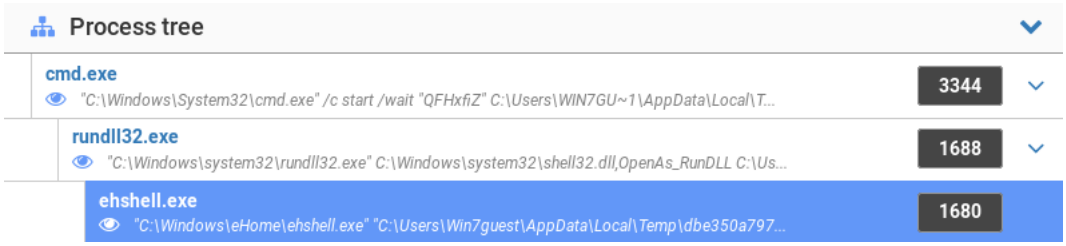


Obr. 4.32: Signatury analýzy prvního dropperu.

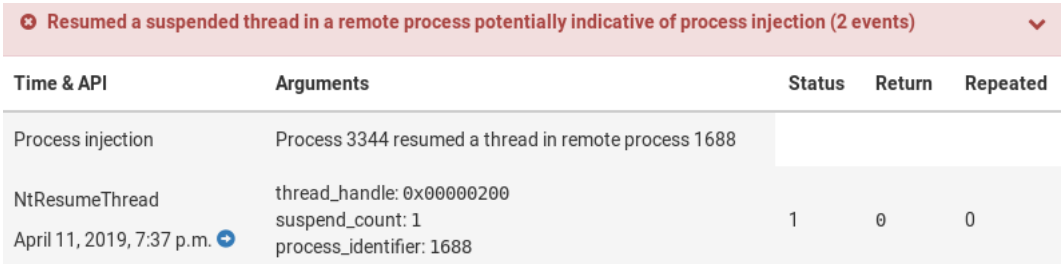
[46].

Dynamická analýza souboru Dropper2

Výsledky této analýzy vykazovaly stejné signatury jako analýza souboru ***Dropper1*** popsaná výše v sekci Dynamická analýza souboru Dropper1(4.4.5). Rozdíl v dosaženém cuckoo skóre byl zapříčiněn detekcí generovaného Internet Control Message

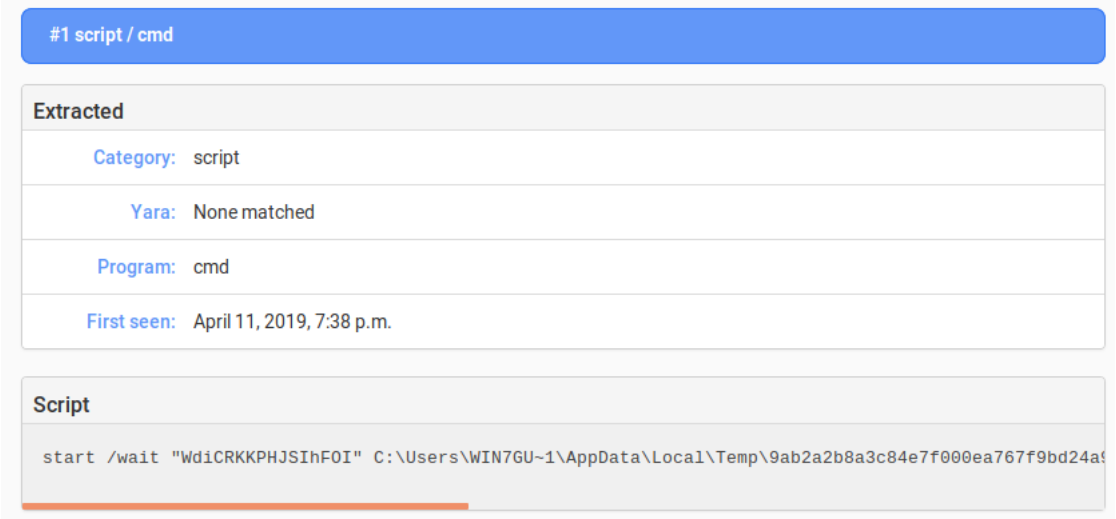


Obr. 4.33: Spuštěné subprocesy dropperu.



Obr. 4.34: Injekce rodičovského procesu dropperu.

Protocol (ICMP) provozu. Na základě wiresharku se jednalo o jednu ICMP zprávu s dotazem na server 8.8.8.8. Oba soubory neprovedly stahování žádných souborů, avšak Cuckoo extrahoval artifact viz obrázek 4.35. Skript je určen k odloženému spuštění souboru v programu cmd.exe.



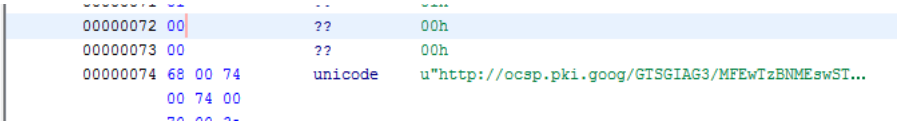
Obr. 4.35: Získaný artifact se skriptem k odloženému spuštění.

Statická analýza dropperů

Jako nástroj pro statickou analýzu jsem použil sadu nástrojů Software Reverse Engineering (SRE) GHIDRA [47]. Při statické analýze jsem odhalil webový odkaz ***Odkaz3*** viz tabulka 4.8. Umístění odkazu je na obrázku 4.36.

Tab. 4.8: Tabulka identifikace odkazu nalezeného programem GHIDRA.

Název v textu	Odkaz3
Webový odkaz	http://ocsp.pki.goog/GTSGIAG3/MFEwTzBNMEswSTAJBgUrDg MCGgUABBT27bBjYjKBmjX2jX WgnQJKEapsrQQUd8K4UJpndnaxLcKG0IO gfqZ%2BuksCEEpXWRnDaZSEY67E8B6coDU%3D
Odkaz získán v analýze číslo	133
Číslo analýzy odkazu	137



Obr. 4.36: Odkaz3 nalezený programem GHIDRA.

Analýza odkazu nalezeného programem GHIDRA

Signatury získané z tohoto souboru nevykazují známky zásadně podezřelého chování a proto jsem přistoupil k analýze stažených souborů. Zde jsem nalezl další soubor, který byl analyzován a v práci ho budu nadále označovat jako ***Dropper3*** viz tabulka 4.9. Soubor vykazoval stejné chování jako vzorky ***Dropper1*** a ***Dropper2***.

Tab. 4.9: Tabulka identifikace souboru v textu označeném jako Dropper3.

Název v textu	Dropper3
Název souboru	22da20d4443e8be3_mfewtzbnmeswstajbgurdgmcgguabbt27bbjyjkbmjx2jxw gnqjkeapsrqqud8k4ujpndnaxlckg0iogfqz+uksceepxwrndazsey67e8b6codu=[1]
SHA256 Hash	22da20d4443e8be369246bce5c91260a08e42fa102ef1bae5dfbfd07bbf8c9d7
Soubor získán v analýze číslo	137
Číslo analýzy souboru	138

4.5 Payload soubory extrahované z paměti

Při procházení a analyzování souborů z paměti analýzy, jsem objevil několik desítek spustitelných souborů. K paměti jednotlivých analýz lze přistoupit díky jednotnému

CWD přistupovat takto `.cuckoo/storage/analyses/číslo_analýzy/memory`. Nalezené soubory byly následně analyzovány Cuckoo sandboxem viz obrázek 4.37. Příklad shrnutí analýzy souboru ***Payload1*** viz tabulka 4.10 je na obrázku 4.38. Tento exe soubor dosáhl nejvyššího skóre a byl nalezen v paměti analýzy souboru ***Dropper3***. Ze shrnutí je patrná vysoká míra entropie dat a také detekce některými antivirovými programy ze souboru VirusTotal. Co se sandboxové analýzy souboru ***Payload1*** týče, nepřinesla již další výsledky. Je to zapříčiněno omezenou možností úspěšně spustit v sandboxovém prostředí takto získané vzorky. Další z nalezených spustitelných souborů, které se mi podařilo najít v paměti důležitých analýz uvádím v tabulce 4.11.

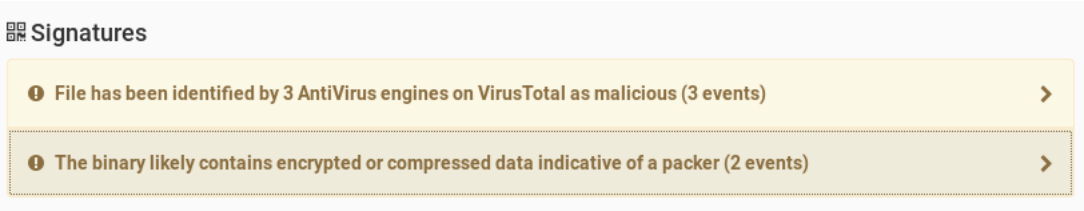
163	2019-05-04 18:41	90b29a7a570ee995abb0b3df9f4ab11e	2964-93550607befac66d.exe_	reported	score: 0
162	2019-05-04 18:41	f5834b1a499ca1b731d5ffa9a85978fa	2964-6492ad1046e40356.exe_	reported	score: 0
161	2019-05-04 18:41	e842d902af9c505857fd59835b4cf9e6	2512-1d998278b090279d.exe_	reported	score: 0
160	2019-05-01 14:07	f6905146e5192a27663851ad70772380	2724-019b6fdeb9b9c8f5.exe_	reported	score: 0.4
159	2019-05-01 14:05	a9a52a508894b7fa2c90090d6441c5f9	2464-858addaa73e0dbca.exe_	reported	score: 0.8
158	2019-05-01 14:03	bce4fea788684677b97d6be5eb18f85b	1564-eef3766387751f9f.exe_	reported	score: 0.4
157	2019-05-01 14:02	f434861aa6dff75c20b84420f20bc205	1564-73e6f5f8b0085f9a.exe_	reported	score: 0.4

Obr. 4.37: Přehlad některých analýz spustitelných souborů.

Tab. 4.10: Tabulka identifikace spustitelného souboru Payload1.

Název v textu	Payload1
Název souboru	2464-858addaa73e0dbca.exe
SHA256 Hash	5e659cda8fbadd6ccffbecbec8b462f06fb89ba2bb7ec7391e364017ecc52ca0
Soubor získán v paměti analýzy	138
Číslo analýzy souboru	159
Detekce VirusTotal	2/65

Jelikož automatická analýza již nebyla dostatečná, pokusil jsem se o statickou analýzu těchto payloadů pomocí programů GHIDRA a xdg64. Soubory jsou však natolik pokročilé, že se mi nepodařilo prolomit jejich zabezpečení a další navazující statická analýza tak pro mne nebyla možná. Informace o těchto finálních payloadech jsem získal na základě nahrání a publikace na servery `virustotal.com` a `hybrid-analysis.com` [22] [48]. O publikaci a získaných informacích budu psát na konci této práce.



Obr. 4.38: Signatury analyzovaných binárních souborů.

Tab. 4.11: Tabulka identifikace extrahovaných souborů z paměti analýz.

Název	SHA256 Hash	Odkaz získán v paměti analýzy	Detekce VirusTotal
612-567aced7146477cf.exe	b484ab198a1d7c9aa76c7cebba5384e2a6f 05f6a567bd0a500deae696e97dd8c	134	4/56
2724-019b6fdeb9b9c8f5.exe	a4df3e39b402d5744cc207adaeb9616f66 80a67a757342316e6587517483f138	138	4/72
1564-73e6f5f8b0085f9a.exe	daa75d2f53423f01b0787244b1594311b2 5cab3d1b67d3a00784b694b2fa80a8	138	3/70
1564-eef3766387751f9f.exe	19d60ee2ac8c8370f7a3891866873a2fe0 c4b8683785339e8b7f8713f91b1654	138	3/71
1400-24e4f90d0b887999.exe	9a1c29f4f724074910c5c19c075dcf0215 8bd2fe1bffefae35e82bbbc37718b0	137	2/72
2144-16903147d9f5ec03.exe	540d9295ad8d9683185a477dde5b983ef1 35877242840a571a05330829ed0b90	137	1/67
2144-ba33d376da599dd8.exe	9a1c29f4f724074910c5c19c075dcf0215 8bd2fe1bffefae35e82bbbc37718b0	137	2/72
3940-6166973527b7df61.exe	b1fc6b9400e02183381fd911b57a4aa2f6 0e9bb7a18f82d9c56c507d67dc04d2	134	4/73
2860-af3f9c6105d4739a.exe	8ae2ee5f5de356f839821e7f71547d205c 2f8dd8cc6c585396f4b6b4f12eb5a5	134	4/65
2860-29c60f998b7197b4.exe	840b0bb821d9517d9b15e275bb74be9a4e 487797f9063c550e5cb6eaa095acfa	134	4/71
1312-edf78ecdecb49524.exe	04bf01b4d3d473b649a3edfeaced4dd9e3 36df3c82f55e1d1e44ac0750083e3c	134	4/51
3344-a43488fc73fe3347.exe	53e3255b5e6b3e57408f852e88d6b5face d1940d69719c131ea364da73cf617b	133	4/66
1688-aef7eadd55d1c883.exe	611185bd9a14023365ffb7431ca3cdf5bd d076d10f5c76ac322ca540741e5620	133	3/73
1688-6167cf3fcc89b544.exe	5d2597db9a3416f0032acc27e82c6ee656 d5310e078dd964ae8ef3881626c498	133	3/72
1680-493330dbe971f2bf.exe	f8ce718ac69ad7cf4cfebeaeeb0018e5c1 024e97066776807a5d51e0c41127e9	133	4/51
1964-4bff68923b59be26.exe	f09655426b660ee271cbf5d16fbee0f403 0bbce272020a3a1df5d1a66ccafffc	132	2/71
692-a91d95ca6886a5ce.exe	7617f860b53691dbe83396efec9ce227c1d dd68c54aa25a00502f443c35ccc3d	132	2/70
692-6667239c73b56e7d.exe	610ed520ebf44add8574a2d9104866eff03 ba7fbcd1217e66720b52cba2309d2	132	2/61
2436-deaff009c4845769.exe	61c627a660c8cdd8c2e6bb458591202d4b aab8619716347f7b13bd95b9a57025	104	2/70
2736-4cda9cf7fd495c2f.exe	7e1580d5fdf88094665a3a093dfe66e431 ba69fd5b6b829698053337ccade6e7	104	2/70
2736-a49cf7c079bb95e7.exe	f49f24d048753097dd67cfea598e8ce8ef f4df80bf2492adf785175eca6003b2	104	2/65

4.6 Publikace souborů

Vzhledem k pokročilým vyhýbavým vlastnostem většiny nalezených vzorků jsem veškeré zajímavé vzorky nahrál na servery `virustotal.com` a `hybrid-analysis.com` [22] [48].

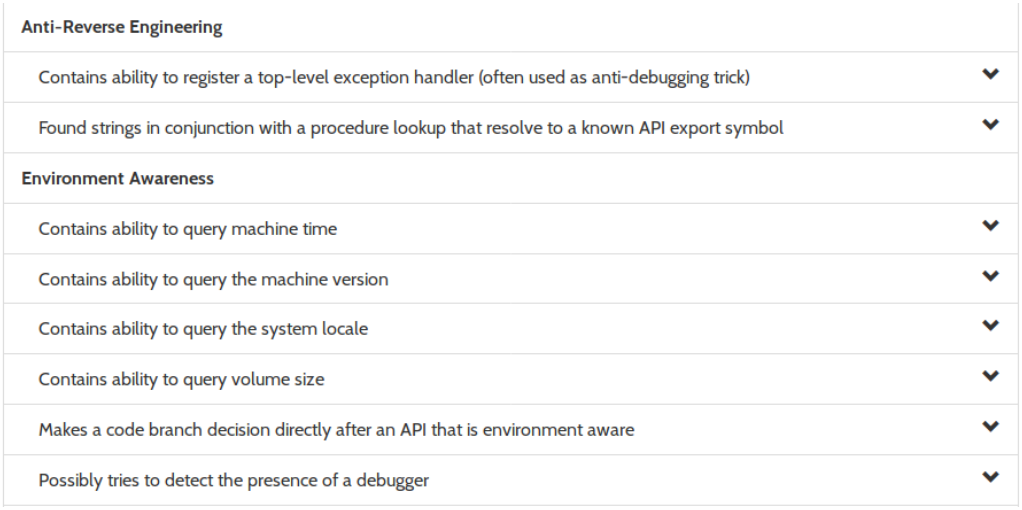
Soubory typu dropper

Na úvod jsem se pokusil ověřit mnou zjištěné chování vzorků ***Dropper1***, ***Dropper2*** a ***Dropper3*** viz tabulky 4.6 4.7 4.9. Toho se mi podařilo docílit za pomoci Falcon sandboxu využitém na stránkách `hybrid-analysis.com` [48]. Problémem analýzy těchto souborů je chybějící typ souboru tzn. v rámci analýzy Cuckoo sandboxem, byl využit balíček „Generic“. Ten soubory typu „raw data“ spouští v programu cmd.exe. Takovou možnost však online analýza za pomocí Falcon sandboxu neposkytuje a proto bylo nutné vyzkoušet různé spustitelné koncovky pro systém Windows. Koncovka umožňující správné provedení analýzy ve všech třech případech byla .bat.

Stejně jako v rámci mnou provedených analýz pomocí Cuckoo, byly soubory svým chováním velice podobné i v prostředí Falcon sandbox. Výsledkem bylo skóre 35/100 a označení „suspicious“ u všech tří souborů. Náhled indikátorů nalezených Flacon sandboxem značící schopnost vzorků detekovat virtuální prostředí a testování přítomnosti debuggeru je na obrázku 4.39. Nalezení podrobných informací o analýze je také možné skrze hash souboru uvedený v patřičné tabulce. Co se nahrání dropperů na porátl VirusTotal týče, ve všech třech případech nebyl soubor shledán nebezpečným žádným z přítomných antivirových nástrojů. Dle poznatků z Falcon sandboxu usuzuji, že v rámci nahrání a testování souborů jednotlivými nástroji mohl nastat problém s typem souboru a patřičnou koncovkou. Již při analýze Falcon sandboxem bylo spuštění kódu neúplné a ani přes aktivní komunikaci s podporou stránky Hybrid-analysis se mi nepodařilo provést úspěšné a plné spuštění kódu. Proto jsem tyto vzorky okomentoval dle oficiálního doporučení komentování vzorků a to `#malware` spolu s mnou zjištěnými informacemi. Webové odkazy k těmto publikovaným souborům jsou v příloze „**A Obsah přiloženého CD**“ ve složce Odkazy_publikace.

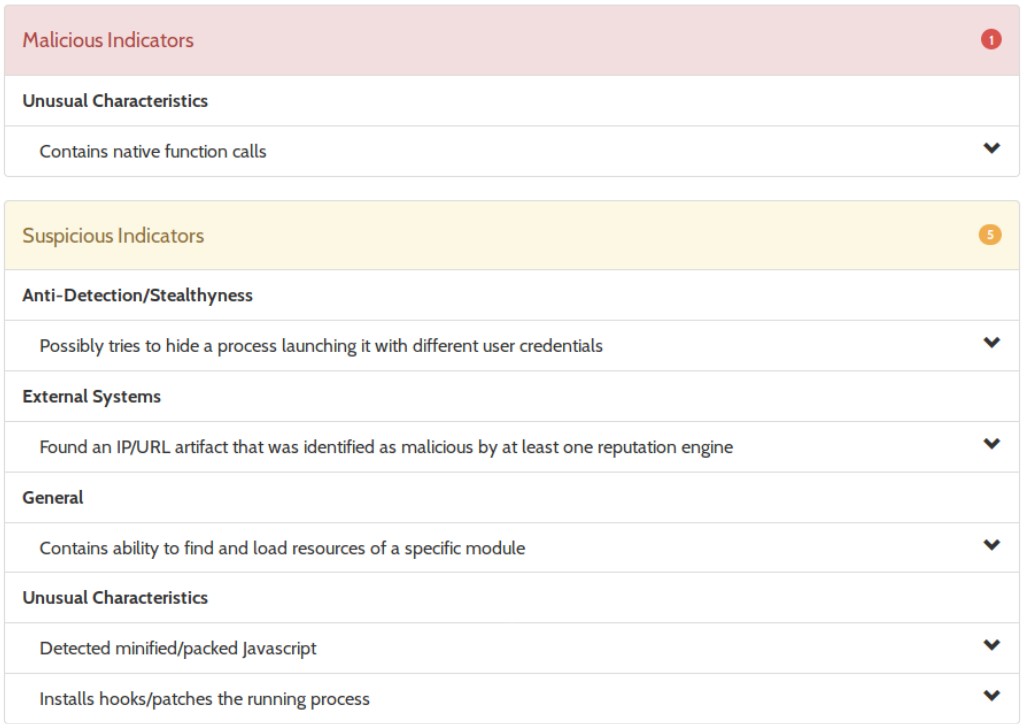
JavaScript1

V případě analýzy Falcon sandbox bylo vyhodnoceno skóre 42/100 a soubor byl označen jako „suspicious“. Indikátory byly oproti mnou zjištěným signaturám v prostředí Cuckoo obsáhlejší a zajímavější viz obrázek 4.40. Na něm je patrná například položka `Installs hooks/patches the running process`, která upozorňuje na zápis do virtuální adresy v paměti modulu user32.dll. Dále je zde polož-



Obr. 4.39: Indikace nalezené Falcon sandboxem potvrzující zjištění Cuckoo sandboxu ohledně souborů typu dropper [48].

kou `Detected minified/packed javascript` upozorněno na vysoký poměr syntaxe oproti popisovačům a jedním indikátorem v seznamu škodlivých indikátorů `Contains native function calls`.



Obr. 4.40: Indikace nalezené Falcon sandboxem ohledně souboru JavaScript1[48].

Antivirové nástroje serveru VirusTotal neoznačily soubor ***JavaScript1*** jako škodlivý. Proto jsem vzorek okomentoval jako „obfuskovaný“ JavaScript obsahující škodlivý URL odkaz a přidal webovou adresu analýzy serveru hybrid-analysis.com. Webový odkazy k tomuto publikovanému souboru je v příloze „**A Obsah přiloženého CD**“ ve složce Odkazy_publikace.

Soubory typu payload

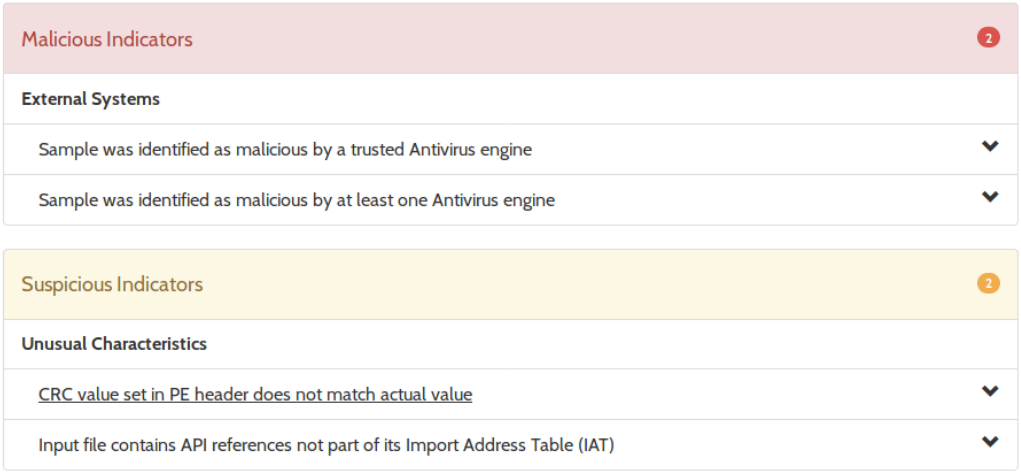
Jako nejnebezpečnější vzorek se ukázal soubor označený jako ***Payload2*** viz tabulka 4.12. Co se Falcon sandboxu týče, měl s analýzou souboru stejné problémy jako já s využitím Cuckoo. Tento poznatek jenom podporuje moji prvotní myšlenku, že se jedná o pokročilé vzorky a jejich další analýza přesahuje rámec této práce. S přihlédnutím k potížím se spuštěním payloadů v sandboxovém prostředí uvádím indikátory analýzy ***Payload2*** s nejvyšším počtem detekcí ke dni 19.5.2019 na obrázku 4.41. Zde je patrné, že jako škodlivé indikátory analýza považuje informace získané právě z analýz portálu VirusTotal atp. Dále je možné vidět indikátor `CRC value set in PE header does not match actual value`, který varuje před nižším kontrolním součtem udávaným v hlavičce souboru oproti reálnému kontrolnímu součtu.

Tab. 4.12: Tabulka identifikace spustitelného souboru Payload2.

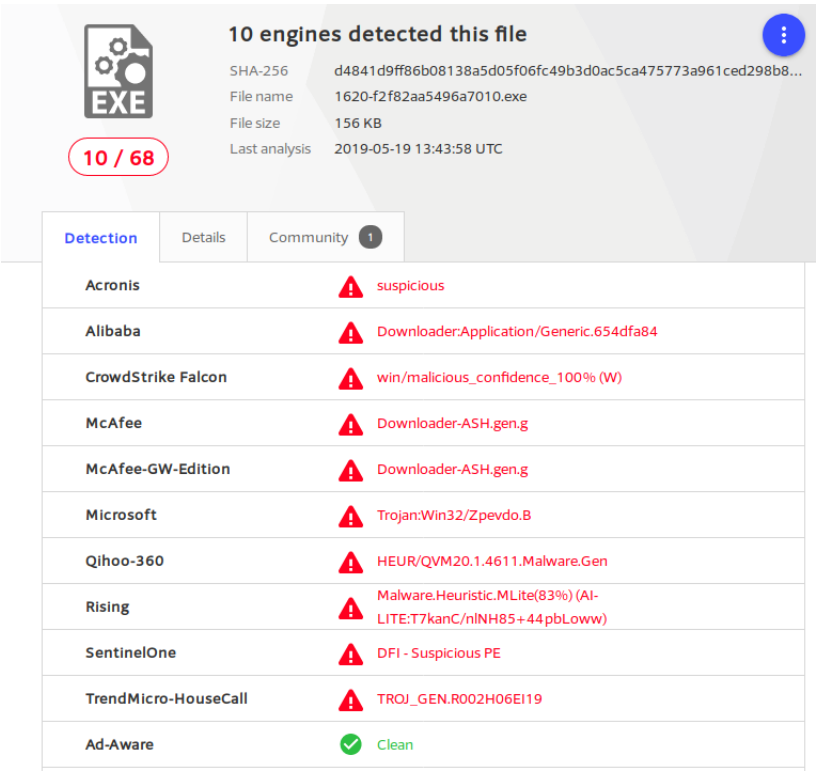
Název v textu	Payload2
Název souboru	1620-f2f82aa5496a7010.exe
SHA256 Hash	d4841d9ff86b08138a5d05f06fc49b3d0ac5ca475773a961ced298b87962f59d
Soubor získán v paměti analýzy	129
Číslo analýzy souboru	142
Detekce VirusTotal	10/68

Jak již bylo zmíněno výše, tento soubor na portálu VirusTotal zaznamenal 10/68 detekcí ke dni 19.5.2019 viz obrázek 4.42. Vzhledem k obtížnostem spojeným s detekcí souboru v sandboxovém prostředí, jsou informace o detekci pomocí jednotlivých antivirových nástrojů webu VirusTotal nejsměrodatnější za účelem identifikace souborů.

Co se finálních payloadů týče, všechny byly okomentovány na portálu VirusTotal jako `#drive-by-download`. vzhledem k proměnlivosti počtu detekcí, jsou veškeré detekční poměry uvedeny v tabulce 4.11 u příslušných exe souborů k datu 19.5.2019. Webové odkazy k těmto publikovaným souborům jsou v příloze „**A Obsah přiloženého CD**“ ve složce Odkazy_publikace.



Obr. 4.41: Indikace nalezené Falcon sandboxem ohledně souboru Payload2[48].



Obr. 4.42: Detekce souboru Payload2 serverem Virus Total[22].

5 Závěr

Výsledkem této práce je navržené sofwarové řešení pro analýzu škodlivého kódu a to prostředím Cuckoo sandbox s potřebnými doplňujícími nástroji pro statickou analýzu jako Process Explorer, Process Monitor. Prostředí je instalováno v souladu s požadavkem na co možná nejvíce izolované prostředí. Dále je výsledkem analýza chování vybraného škodlivého kódu generovaného nástrojem Pupy. Součástí výstupu práce je také seznámení s teoretickými principy potřebnými pro sandboxing a odhalování dopadů škodlivých kódů na systém za pomoci statických a dynamických analýz. Při zadání práce bylo uvažováno o navržení hardwarového prostředí pro sandbox, na základě konzultací však bylo od tohoto procesu upuštěno. Přesto je součástí práce alespoň popis hardwaru na kterém byla práce provedena.

Dále je výsledkem této práce nalezení několika desítek podezřelých vzorků odhalených v praktické analýze webového prostředí se zaměřením na malvertisingové kampaně. V rámci tohoto snažení je také popsáno několik principů a technik pro analýzu různých typů vzorků. Vlastním přínosem bakalářské práce je také výsledná analýza a okomentování podezřelých vzorků, které jsou uvedeny v příloze „**A Obsah přiloženého CD**“ ve složce Odkazy_publikace.

Veškeré stanovené cíle bakalářské práce byly splněny.

Literatura

[1] JIROVSKÝ, Václav. Kybernetická kriminalita: nejen o hackingu, crackingu, virech a trojských koních bez tajemství. Praha: Grada, 2007. ISBN 978-80-247-1561-2.

[2] DISTLER, Dennis. Malware Analysis: An Introduction. SANS Institute [online]. ©2000-2018, 14 Dec 2007, , 1-68 [cit. 2018-12-01]. Dostupné z: <https://www.sans.org/reading-room/whitepapers/malicious/malware-analysis-introduction-2103>

[3] PERRIN, Chad. Hacker vs. cracker. In: Techrepublic [online]. 2009-04-17 [cit. 2018-11-05]. Dostupné z URL:<https://www.techrepublic.com/blog/it-security/hacker-vs-cracker/>

[4] GORDON, Sarah. Fighting Spyware and Adware in the Enterprise. EDPACS. 2005, 32(12), 14-18. DOI: 10.1201/1079/45242.32.12.20050601/88294.2. ISSN 0736-6981. Dostupné také z: <http://www.tandfonline.com/doi/abs/10.1201/1079/45242.32.12.20050601/88294.2>

[5] ABUZAID, Areej Mustafa, Madihah Mohd SAUDI, Bachok M TAIB a Zul Hilmi ABDULLAH. An Efficient Trojan Horse Classification (ETC). IJCSI International Journal of Computer Science Issues. www.IJCSI.org, 2013, 2013(No 3), 96-104. ISSN 1694-0784.

[6] SGANDURRA, Daniele, Luis MUÑOZ-GONZÁLEZ, Rabih MOHSEN a Emil C. LUPU. Automated Dynamic Analysis of Ransomware:Benefits, Limitations and use for Detection. Department of Computing, Imperial College London180 Queen's Gate, SW7 2AZ, London, UK, 2016.

[7] LOBO, D., P. WATTERS a XINWEN WU. RBACS: Rootkit Behavioral Analysis and Classification System. 2010 Third International Conference on Knowledge Discovery and Data Mining. IEEE, 2010, 2010, , 75-80. DOI: 10.1109/WKDD.2010.23. ISBN 978-1-4244-5397-9. Dostupné také z: <http://ieeexplore.ieee.org/document/5432724/>

[8] Bad Rabbit ransomware: why we should be worried about this new threat: 4. Dec 2017. Verdict [online]. ©2018 [cit. 2018-12-01]. Dostupné z: <https://www.verdict.co.uk/bad-rabbit-ransomware-new-threat/>

[9] Malware Examples and how to remove them. Comodo Antivirus [online]. ©2019, 17 Jul 2018 [cit. 2019-05-19]. Dostupné z: <https://antivirus.comodo.com/blog/comodo-news/malware-examples-and-their-removal/>

[10] CHOW, Sherman S.M., Lucas C.K. HUI, S.M. YIU, K.P. CHOW a Richard W.C. LUI. A generic anti-spyware solution by access control list at kernel level. Journal of Systems and Software. 2005, 75(1-2), 227-234. DOI: 10.1016/j.jss.2004.05.027. ISSN 01641212. Dostupné také z: <https://linkinghub.elsevier.com/retrieve/pii/S0164121204000949>

[11] OBERHEIDE, Jon, Evan COOKE a Farnam JAHANIAN. CloudAV: N-Version Antivirus in the Network Cloud. Proceedings of the 17th USENIX Security Symposium. San Jose, CA, USA, 2008, , 91-106.

[12] Co je RTB a jak funguje?. Marketup [online]. 13 Oct 2013 [cit. 2019-05-17]. Dostupné z: <http://www.marketup.cz/cs/blog/co-je-rtb-a-jak-funguje>

[13] DWYER, Catherine a Ameet KANGURI. Journal of information systems applied research. Chicago, Ill.: Association of Information Technology Professionals, Education Special Interest Group, 2017. ISBN 1946-1836. ISSN 1946-1836.

[14] SEGURA, Jérôme. Malvertising Hits DailyMotion, Serves Up Angler EK. Malwarebytes Labs [online]. ©2019, 7 Dec 2015 [cit. 2019-05-17]. Dostupné z: <https://blog.malwarebytes.com/threat-analysis/2015/12/malvertising-hits-dailymotion-serves-up-angler-ek/>

[15] MOSER, Andreas, Christopher KRUEGEL a Engin KIRDA. IEEE. Limits of Static Analysis for Malware Detection. 2007, 10 s. ISBN 978-0-7695-3060-4. ISSN 1063-9527. Dostupné také z: <https://auto.tuwien.ac.at/~chris/research/doc/acsac07_limits.pdf>

[16] JavaScript Obfuscator Tool [online]. [cit. 2019-05-18]. Dostupné z: <https://obfuscator.io/>

[17] JUN KWON, Bum, Jayanta MONDAL a Jiyong JANG. The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics. Denver, Colorado, USA, 2015. ISBN 978-1-4503-3832-5. Dostupné také z: <http://www.cs.umd.edu/~jayanta/papers/ccs15.pdf>

[18] GOGAN, Marcell. Sandbox-Evading Malware Are Coming: 7 Most Recent Attacks. Hakin9 [online]. ©2013, 21 Aug 2018 [cit. 2019-05-18]. Dostupné z: <https://hakin9.org/sandbox-evading-malware-are-coming-7-most-recent-attacks/>

[19] WATSON, Jon. What is sandboxing and how to sandbox a program. In: Comparitech [online]. ©2018, 2 Mar 2018 [cit. 2018-12-01]. Dostupné z: <https://www.comparitech.com/blog/information-security/what-is-sandboxing/

[20] NUSBAUM, Scott. Malware Analysis is for the (Cuckoo) Birds. TrustedSec [online]. ©2018, 18 May 2018 [cit. 2018-12-01]. Dostupné z: <https://www.trustedsec.com/2018/05/malware-cuckoo-1/>

[21] SIKORSKI, Michael a Andrew HONIG. Practical malware analysis: the hands-on guide to dissecting malicious software. San Francisco: No Starch Press, c2012. ISBN 978-1-59327-290-6.

[22] VirusTotal [online]. [cit. 2018-12-01]. Dostupné z: <https://www.virustotal.com/#/home/upload>

[23] Malware Hash Query. Vicheck [online]. [cit. 2018-12-01]. Dostupné z: <https://www.vicheck.ca/hashquery.php>

[24] INetSim: Internet Services Simulation Suite [online]. ©2007-2018 [cit. 2018-12-01]. Dostupné z: <https://www.inetsim.org/index.html

[25] AMALI, Warunika. Cuckoo Sandbox Installation Guide. Medium [online]. 8 Jul 2017 [cit. 2018-12-01]. Dostupné z: <https://medium.com/@warunikaamali/cuckoo-sandbox-installation-guide-d7a09bd4ee1f>

[26] Venv — Creation of virtual environments [online]. ©2001-2018 [cit. 2018-12-01]. Dostupné z: <https://docs.python.org/3/library/venv.html>

[27] Virtualenv [online]. ©Copyright2007-2018 [cit. 2018-12-01]. Dostupné z: <https://virtualenv.pypa.io/en/latest/#>

[28] Cuckoo Sandbox Book [online]. ©2010-2018 [cit. 2018-12-01]. Dostupné z: <https://cuckoo.sh/docs/index.html>

[29] Linode. Linux Users and Groups. Https://www.linode.com/ [online]. ©2018, 23 Mar 2017 [cit. 2018-12-01]. Dostupné z: <https://www.linode.com/docs/tools-reference/linux-users-and-groups/>

[30] The GNU Nano [online]. 2018 [cit. 2018-12-01]. Dostupné z: <https://www.nano-editor.org/docs.php>

[31] Pupy. GitHub [online]. ©2018 [cit. 2018-12-01]. Dostupné z: <https://github.com/n1nj4sec/pupy>

[32] Stažení Mozilla Firefox. Mozilla.cz [online]. ©1998-2018 [cit. 2018-12-02]. Dostupné z: <https://www.mozilla.org/cs/firefox/download/thanks/?from=mozillacz&scene=2>

[33] BOUŠKA, Petr. TCP/IP - adresy, masky, subnety a výpočty. Samuraj-cz [online]. ©2005-2018, 11 Aug 2018 [cit. 2018-12-02]. Dostupné z: <https://www.samuraj-cz.com/clanek/tcpip-adresy-masky-subnety-a-vypocty/>

[34] Cuckoo Sandbox [online]. [cit. 2019-04-22]. Dostupné z: <https://github.com/cuckoosandbox>

[35] MUELLER, Lance. ForensicKB. File Entropy explained [online]. 2013 [cit. 2019-05-23]. Dostupné z: <http://www.forensickb.com/>

[36] NetBIOS Over TCP/IP: 18 Jul 2012. Microsoft Docs [online]. [cit. 2018-12-08]. Dostupné z: <https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-2000-server/cc940063(v=technet.10)>

[37] NirSoft [online]. [cit. 2019-05-20]. Dostupné z: <https://www.nirsoft.net/>

[38] OLUWAFEMI, Kayode. A Beginners Guide to using apt-get commands in Linux(Ubuntu). : codeburst.io [online]. 18 Mar 2018 [cit. 2019-05-16]. Dostupné z: <https://codeburst.io/a-beginners-guide-to-using-apt-get-commands-in-linux-ubuntu-d5f102a56fc4>

[39] OpenSC [online]. [cit. 2019-05-16]. Dostupné z: <https://www.opensc.io>

[40] Hack Forums [online]. [cit. 2019-05-16]. Dostupné z: <https://www.hackforums.net>

[41] DNSQUERY [online]. ©2008-2019 [cit. 2019-05-04]. Dostupné z: <https://dnsquery.org/>

[42] X64dbg [online]. [cit. 2019-05-09]. Dostupné z: <https://x64dbg.com/#start>

[43] OllyDbg [online]. 2014 [cit. 2019-05-09]. Dostupné z: <http://www.ollydbg.de/>

[44] Online JavaScript Beautifier [online]. [cit. 2019-05-09]. Dostupné z: <https://beautifier.io/>

[45] Debugging Functions: IsDebuggerPresent function. Microsoft [online]. ©2019, © 2019 [cit. 2019-05-11]. Dostupné z: <https://msdn.microsoft.com/en-us/library/windows/desktop/ms680345(v=vs.85).aspx>

[46] How to obtain a handle to any process with SeDebugPrivilege. Microsoft [online]. ©2019, © 2019 [cit. 2019-05-11]. Dostupné z: <https://support.microsoft.com/en-us/help/131065/how-to-obtain-a-handle-to-any-process-with-sedebugprivilege>

[47] GHIDRA [online]. [cit. 2019-05-11]. Dostupné z: <https://ghidra-sre.org/>

[48] Hybrid analysis [online]. ©2019 [cit. 2019-05-14]. Dostupné z: <https://www.hybrid-analysis.com/>

Seznam symbolů, veličin a zkratek

Adware	Advertising-supported software
API	Application Programming Interface
APT	Advanced Package Tool
AVS	antivirus software
Botnet	Robot network
CWD	Cuckoo Working Directory
DDOS	distributed denial of service
DLL	Dynamic-link library
DNS	Domain Name System
exe	executable
GUI	Graphical User Interface
HIPS	host intrusion prevention system
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
KVM	Kernel-based Virtual Machine
PID	process identifier
POP3	Post Office Protocol
PPI	Pay Per Install
RAT	Remote Administration Tool
RTB	Real Time Bidding
SRE	Software Reverse Engineering
SSL	Secure Sockets Layer
TFTP	Trivial File Transfer Protocol
Trojan	Trojan Horse
UDP	User Datagram Protocol
URL	Uniform Resource Locator
virtualenv	Python Virtual Enviromment

Seznam příloh

A Obsah přiloženého CD

```
---------------------- Přílohy
Odkazy_publikace
    JavaScript
        Odkaz_JavaScript_VirustTotal.txt
    Payloady
        Odkazy_Payloady_VirusTotal.txt
    Droppery
        Odkazy_Droppery_VirusTotal.txt
Extrahovany_JavaScript
    6cf156d528af2d1791543fa0eddc3e1cef2f522a79bdc8ce47ba99440755a
    1eb_6cf156d528af2d17_gpt[1].js
Extrahovane_Payloady
    104
        2736-4cda9cf7fd495c2f.exe
        2736-a49cf7c079bb95e7.exe
        2436-deaff009c4845769.exe
    132
        1964-4bff68923b59be26.exe
        692-6667239c73b56e7d.exe
        692-a91d95ca6886a5ce.exe
    133
        1680-493330dbe971f2bf.exe
        1688-6167cf3fcc89b544.exe
        3344-a43488fc73fe3347.exe
        1688-aef7eadd55d1c883.exe
    134
        2860-af3f9c6105d4739a.exe
        3940-6166973527b7df61.exe
        1312-edf78ecdecb49524.exe
        2860-29c60f998b7197b4.exe
    129
        1620-f2f82aa5496a7010.exe
    138
        2464-858addaa73e0dbca.exe
        2724-019b6fdeb9b9c8f5.exe
        1564-eef3766387751f9f.exe
        1564-73e6f5f8b0085f9a.exe
    137
        2144-ba33d376da599dd8.exe
        1400-24e4f90d0b887999.exe
        2144-16903147d9f5ec03.exe
Extrahovane_Droppery
```

```
    ├── 9ab2a2b8a3c84e7f000ea767f9bd24a9c2787c30982e470890d0b3682bcb2
    │   dc9_9ab2a2b8a3c84e7f_f5f320a94d4d2b4465d8f17e2bb2d351_397e41a
    │   2f252913b147eaf3fa1644be8
    ├── 22da20d4443e8be369246bce5c91260a08e42fa102ef1bae5dfbfd07bbf8c
    │   9d7_22da20d4443e8be3_mfewtzbnmeswstajbgurdgmcgguabbt27bbjyjkb
    │   mjx2jxwgnqjkeapsrqqud8k4ujpndnaxlckg0iogfqz+uksceepxwrndazsey
    │   67e8b6codu=[1]
    └── dbe350a797d3fd7d6ad5d0e1bad1a6458eba8c8f78b418778792db8886ab4
        cf8_dbe350a797d3fd7d_f5f320a94d4d2b4465d8f17e2bb2d351_397e41a
        2f252913b147eaf3fa1644be8
├── Binarni_soubor_programu_pupy
│   └── pupyx64.f3krDi.exe
└── readme.txt
```